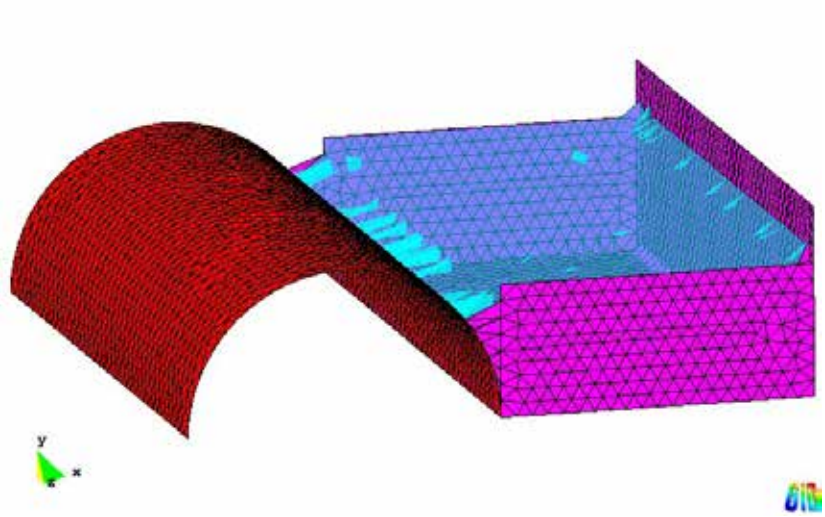
Lagrangian FE Methods for Coupled Problems in Fluid Mechanics

P. Ryzhakov
E. Oñate
R. Rossi
S. R. Idelsohn



Monograph CIMNE Nº-121, November 2010

Lagrangian FE Methods for Coupled Problems in Fluid Mechanics

P. Ryzhakov
E. Oñate
R. Rossi
S. R. Idelsohn

Monograph CIMNE Nº-121, November 2010

International Center for Numerical Methods in Engineering
Gran Capitán s/n, 08034 Barcelona, Spain

INTERNATIONAL CENTER FOR NUMERICAL METHODS IN ENGINEERING
Edificio C1, Campus Norte UPC
Gran Capitán s/n
08034 Barcelona, Spain
www.cimne.upc.es

First edition: November 2010

LAGRANGIAN FE METHODS FOR COUPLED PROBLEMS IN FLUID MECHANICS
Monograph CIMNE M121

ISBN: 978-84-96736-97-9

Depósito legal: B-41230-2010

Summary

Lagrangian finite element methods emerged in fluid dynamics when the deficiencies of the Eulerian methods in treating free surface flows (or generally domains undergoing large shape deformations) were faced. Their advantage relies upon natural tracking of boundaries and interfaces, a feature particularly important for interaction problems. Another attractive feature is the absence of the convective term in the fluid momentum equations written in the Lagrangian framework resulting in a symmetric discrete system matrix, an important feature in case iterative solvers are utilized. Unfortunately, the lack of the control over the mesh distortions is a major drawback of Lagrangian methods. In order to overcome this, a Lagrangian method must be equipped with an efficient re-meshing tool.

This work aims at developing formulations and algorithms where maximum advantage of using Lagrangian finite element fluid formulations can be taken. In particular we concentrate our attention at fluid-structure interaction and thermally coupled applications, most of which originate from practical "real-life" problems. Two fundamental options are investigated - coupling two Lagrangian formulations (e.g. Lagrangian fluid and Lagrangian structure) and coupling the Lagrangian and Eulerian fluid formulations.

In the first part of this work the basic concepts of the Lagrangian fluids, the so-called Particle Finite Element Method (PFEM) [1], [2] are presented. These include nodal variable storage, mesh re-construction using Delaunay triangulation/tetrahedralization and alpha shape-based method for identification of the computational domain boundaries. This shall serve as a general basis for all the further developments of this work.

Next we show how an incompressible Lagrangian fluid can be used in a partitioned fluid-structure interaction context. We present an improved Dirichlet-Neumann strategy for coupling the incompressible Lagrangian fluid with a rigid body. This is finally applied to an industrial problem dealing with the sea-landing of a satellite capsule.

In the following, an extension of the method is proposed to allow dealing with fluid-structure problems involving general flexible structures. The method developed takes advantage of the symmetry of the discrete system matrix and by introducing a slight fluid compressibility allows to treat the fluid-structure interaction problem efficiently in a monolithic way. Thus, maximum benefit from using a similar description for both the fluid (updated Lagrangian) and the solid (total Lagrangian) is taken. We show next that the developed monolithic approach is particularly useful for modeling the interaction with light-weight structures. The validation of the method is done by means of

comparison with experimental results and with a number of different methods found in literature.

The second part of this work aims at coupling Lagrangian and Eulerian fluid formulations. The application area is the modeling of polymers under fire conditions. This kind of problem consists of modeling the two subsystems (namely the polymer and the surrounding air) and their thermo-mechanical interaction. A compressible fluid formulation based on the Eulerian description is used for modeling the air, whereas a Lagrangian description is used for the polymer. For the surrounding air we develop a model based upon the compressible Navier-Stokes equations. Such choice is dictated by the presence of high temperature gradients in the problem of interest, which precludes the utilization of the Boussinesq approximation. The formulation is restricted to the sub-sonic flow regime, meeting the requirement of the problem of interest. The mechanical interaction of the subsystems is modeled by means of a one-way coupling, where the polymer velocities are imposed on the interface elements of the Eulerian mesh in a weak way. Thermal interaction is treated by means of the energy equation solved on the Eulerian mesh, containing thermal properties of both the subsystems, namely air and polymer. The developments of the second part of this work do not pretend to be by any means exhaustive; for instance, radiation and chemical reaction phenomena are not considered. Rather we make the first step in the direction of modeling the complicated thermo-mechanical problem and provide a general framework that in the future can be enriched with a more detailed and sophisticated models. However this would affect only the individual modules, preserving the overall architecture of the solution procedure unchanged.

Each chapter concludes with the example section that includes both the validation tests and/or applications to the real-life problems. The final chapter highlights the achievements of the work and defines the future lines of research that naturally evolve from the results of this work.

Resumen

Los métodos de elementos finitos lagrangianos surgieron en el ámbito de la dinámica de fluidos cuando se detectaron las deficiencias de los métodos eulerianos en su aplicación a problemas de superficies libres. La ventaja de los métodos lagrangianos consiste en su capacidad natural para seguir las interfaces y las fronteras, una característica particularmente importante para problemas de interacción. Otra característica atractiva, es la ausencia de los términos convectivos en las ecuaciónes de conservación de la cantidad de movimento cuando estas se escriben en forma lagrangiana, resultando en un sistema de ecuaciones discretas simétrico. Este hecho define una ventaja importante cuando se utilizan solvers iterativos para su resolución. Desafortunadamente, las graves distorsiones en la malla encontradas en los problemas de fluidos, constituyen la mayor desventaja de los métodos lagrangianos obligando a un remallado frecuente. El objetivo principal de este trabajo es desarrollar formulaciones y algoritmos con los que se pueda aprovechar al máximo las ventajas de los métodos lagrangianos. En particular, concentramos nuestra atención en la interacción fluido-estructura y en aplicaciones térmicamente acopladas. Para ello, se han desarrollado dos estratégias conceptualmente diferentes: 1) acoplamiento de dos formulaciones lagrangianas (fluido y sólido lagrangiano), y 2) acoplamiento de un fluido lagrangiano con uno euleriano.

En la primera parte de este trabajo se presentan los conceptos básicos de un tipo específico de método lagrangiano: el Particle Finite Element Method (PFEM). Esto incluye el almacenaje de las variables en los nodos, remallado utilizando triangulación de Delaunay y el método de alpha-shape para detectar fronteras. Los anteriores conceptos sirven de base general para las fomulaciones desarrolladas y presentadas en este trabajo.

Después se muestra como se puede aplicar una formulación de fluido lagrangiano incompresible en el contexto de interacción fluido-estructura. Se presenta una ampliación de la estrategia Dirichlet-Neumann para acoplar el fluido lagrangiano incompresible con un cuerpo rígido. Esta estrategia se aplica a la resolución de un problema industrial de amerizaje de un satélite.

Acto seguido, se incluye el desarrollo de una extensión del método PFEM, que permite resolver de forma eficiente los casos de interacción entre un fluido y sólidos flexibles. Para ello, se aprovecha la simetría del sistema de ecuaciones y se introduce una pequeña compresibildad al fluido que permite tratar el problema de forma monolítica. Mostramos que la formulación desarrollada es particularmente ventajosa para problemas interacción con estructuras ligeras. El método desarrollado se ha validado comparando los resultados obtenidos con los existentes en la literatura y datos experimentales disponibles.

La segunda parte del trabajo consiste en acoplar la formulación lagrangiana con la euleriana. Esta estrategia es especialmente beneficiosa para la simulación del derretimiento de polímeros expuestos a situaciones de fuego. La formulación euleriana compresible se utiliza para modelar el aire, mientras que el polímero adopta la formulación lagrangiana. La interaccion mecánica entre los sub-dominios se simula mediante el método de acoplamiento debil en una sola dirreción. Las condiciones de interface se asignan de forma aproximada. Por otro lado, el problema térmico se resuelve sobre la malla euleriana, que conteniene sub-dominios con propiedades tanto del aire como del polímero.

Cada capítulo concluye con una serie de ejemplos, que incluyen tanto la validación como la aplicación de los métodos desarrollados a problemas de la practica. El último capítulo resume los principales desarrollos de la tesis y define las líneas de investigación a seguir en un futuro, que surgen de forma natural a partir de los resultados de este trabajo.

Acknowledgements

Four years of work in the field of numerical methods is a highly non-linear process. It contains many bifurcation and stagnation points, a number of discontinuities. Additionally sometimes it even behaves in a non-deterministic way. The catalyst that keeps this complicated process rolling on is the whip and honey (or as we say in Russian "knut i prianik") that is being provided by the supervisor, whose aim is to lead you to the level at which no more catalyst is necessary and the supervised student starts more or less acting as a sort of perpetum mobile, meaning in other words becoming self-sufficient. If I ever reached this point, the person I should thank with all my heart is Dr. Riccardo Rossi, whose scientific support was as grand as the human one. Riccardo, I really appreciate all the help you've been giving during these long four years!

Next I would like to thank Prof. Eugenio Oñate for a number of things. First of all he was the one who has given that spectacular speech at the Technical University of Munich in 2004 and planted in my mind a seed of interest in the Particle Finite Element Method, which made me come to CIMNE. Later on during the time I have been spending here at CIMNE his sharp practical and precise mind has guided me through the long and winding roads of numerical methods and their industrial applications. Thank you so much, Prof. Oñate!

Prof. Sergio Idelsohn was always there so kind to discuss and debate the PFEM related issues, propose challenging examples and share his great experience in the field. I gained a great number of insights from conversations with him.

Next I would like to thank Prof. Ramon Codina and his former student Dr. Javier Principe for their continuous support and advises regarding Eulerian incompressible and compressible flows.

I very much appreciate the detailed corrections of this work made by Prof. Norberto Nigro. The the comments Prof. Nigro provided have to a great extent facilitated improvement of the manuscript.

My fellow mates Kazem Kamran and Dr. Julio Marti have been a great asset to have next to myself. Those were exactly the conversations with Kazem on the pressure shur and the stabilization techniques that have encouraged me to move from one of the main stagnation points I have faced in these 4 years. Julio has been helping with the articles revision that emerged in this work and was always providing me hints and useful advises. Dr. Adelina Geyer and Julio have been so extremely nice to help me with translations into Spanish language. Thanks a lot to you guys!

Hearty thanks go to Dr. Pooyan Dadvand who never hesitated to coach me in templates of C++ and general programming questions and to Mr. Enrique Escolano for his engagement in resolution of the GiD related and unrelated problems.

Last but not least my gratefulness goes to my parents Boris Ryzhakov and Elena Barbanel who have been supporting me throughout these years in all possible ways. It was my mom who mentioned first the idea of me going to Spain for part of my studies.

Special hearty thanks go to my girlfriend Narges for all her patience and understanding during the final stage of this work. She has been the one who has cheered up in the moments I was down and gloomy and has given me many strategic hints. I dedicate this work to my grandmother Prof. Galina Nikolaevna Kuznetsova, whom I consider an example of a scientific mind. She encouraged me to complete this work.

The financial support received from Marie Curie doctoral grant is acknowledged.

I express my gratitude to all the people who have supported me during these 4 years.

To my grandmother G.N. Kuznetsova

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Brief outlook of FEM in CFD

The Finite Element Method (FEM) enjoyed wide popularity in the field of computational structural dynamics (CSD) since the very moment of its birth in the middle of 20th century [7]. FEM has shown to be a powerful tool for the solution of differential equations on domains of complex geometries. The success of the FEM applied to governing equations of structural dynamics was due to the fact that the resulting discrete equations were characterized by symmetric positive-definite system matrices. It was later shown that Galerkin FEM (that is the classical FEM based upon Galerkin formulation of the method of weighted residuals [8]) leads to symmetric systems when applied to self-adjoint parabolic or elliptic differential equations; the momentum equations for an elastic solid falling into the second category. In this case the Galerkin FEM is optimal with respect to the difference between the approximate and exact solution measured in energy norm. Thus the Galerkin method lead to excellent convergence rates in the sense of approaching the exact solution with decreasing element sizes.

Complications arose from the very beginning when the derived methods were transfered to the field of fluid dynamics. It was found that the mathematical structure of fluid dynamics equations in the presence of convective term leads to unsymmetric generalized stiffness matrices. The solution of convection-dominated flows turned out to exhibit severe oscillations or "wiggles" in the velocity field.

Another problem was faced when treating incompressible flows, causing pressure instability for standard linear velocity-pressure elements. It was discovered that the functional spaces for velocity

and pressure cannot be chosen arbitrary, but must rather satisfy the so-called "inf-sup" condition (see e.g. [9]). This precluded in principle utilization of the equal order pressure-velocity interpolation pairs.

Luckily, the discovery of stabilization methods, originally introduced by Hughes et al in the beginning of 80's [10], [11], [12] and their further developments [13], [14] (to cite just a few) enabled circumventing these shortcomings and opened the way to the successful extension of FEM methodology to the area of computational fluid dynamics (CFD). The idea of stabilization methods consists in modifying the weak form of the problem in order to obtain discrete operators with improved properties (symmetry, positive definiteness etc.), but without violating the consistency (i.e. solutions of the modified problem remain being solutions of the original problem).

Once the fundamental mathematical difficulties were resolved and the general FE basis for the fluids was established, the necessity to simulate challenging real-life problems brought into light new obstacles requiring the development of additional algorithms. In particular, the modeling of **free-surface flows** and the analysis of the **fluid-structure interaction** problem have defined new "hot-spots" in the computational mechanics community. These areas and the associated difficulties are strongly connected with the "kinematic frameworks" for the description of the continuum motion, an issue we briefly address next.

1.2 Lagrangian methods

The **Lagrangian** description of motion is a way of looking at a continuum where the observer follows an individual "particle" belonging to the continuum as it moves in time and space. In contrast, in the **Eulerian** description, the observations are made at a fixed location in space, through which the continuum passes.

In the FE context these two descriptions have additional implications. In Lagrangian FE the computational grid (mesh) follows the continuum in its motion, mesh nodes being permanently attached to the same material points. The material coordinates permit one to define then the reference configuration, in which the continuum equations are written. One distinguishes between the **total Lagrangian** and the **updated Lagrangian** formulations. In the total Lagrangian formulation the governing equations are written in a reference frame that corresponds to the material configuration at the initial time (or the undeformed configuration). Updated Lagrangian formulations imply that the reference configuration is being updated at every step.

In an Eulerian framework the finite element mesh is fixed and the continuum moves and deforms

with respect to the computational grid. In the Eulerian formulation, the mesh nodes are dissociated from the material particles. The convection effect originates from the deformation of the material with respect to the fixed grid. The presence of the convective term defines the main distinguishing feature with respect to the form of the governing equations in Eulerian descriptions.

Usually Lagrangian descriptions are used when treating CSD problems, whereas the Eulerian framework is preferred when CFD problems are considered. This is connected with the fact that the domain deformations usually encountered in fluids are so large, that the associated element deformation becomes unacceptable. This drawback of Lagrangian methods had led to the predominating use of Eulerian descriptions in CFD. Even though there is no doubt that Eulerian methods define the optimal choice for flows in enclosures (vessel compliant flows), they do not provide any natural way for treating free surface flows, where the shape of the computational domain is continuously evolving. Ad-hoc techniques such as the so-called "level-set" [15] and "volume of fluid" [16] methods were developed to facilitate the extension of Eulerian methods into free-surface flow modeling. In case of fluid-structure interaction problem the situation becomes even more involved due to the need for the interface identification, data transfer, interface boundary conditions applications and many more.

An alternative to this relies on applying the Lagrangian framework in conjunction with re-meshing, that would permit one to avoid severe mesh distortion associated with the domain deformation. In the Lagrangian formulation the motion of the individual particles are followed and, consequently, nodes in a finite element mesh can be viewed as moving particles. Hence, the motion of the mesh discretizing the domain naturally determines the free surface evolution. Additionally when adopting the Lagrangian approach only the actual physical domain is discretized, while Eulerian methods require discretization of the entire domain, where the material can potentially appear in the solution of the transient problem.

The advantage of the Lagrangian approach relies on the natural treatment of domain configuration (movement of the free surface) and on the absence of convective term in the momentum equation, which means that no stabilization of convection is required. Nevertheless, re-meshing precludes storing the variables of interest at the Gauss points of elements, as it is done in the classical FEM. This can be overcome by means of storing the variables in the nodes instead, resulting in a method which is a hybrid between the FEM and a mesh-free method. This class of Lagrangian methods was developed by Idelsohn, Oñate and Del Pin in the end of 90s [1], [2] and its extension is the focus of the current work.

Lagrangian methods have also a number of important advantages for the fluid-structure inter-

action problems. Except for natural free surface/interface tracking they simplify application of the interaction boundary conditions. We show in this work the benefit of using an updated Lagrangian fluid in conjunction with the standard total Lagrangian structure for definition of a monolithic fluid-structure system. This combined approach has a number of advantages for the interaction between fluids and light-weight structures, a field where many conventional approaches fail.

Lagrangian methods are also found to be beneficial for modeling thermally coupled problems, where fluid properties can change drastically. The intrinsic "transport" of a property to a new position in a Lagrangian fluid is an important advantage in comparison with Eulerian approaches, where for each property a transport equation needs to be solved.

It is worth mentioning that there exist other classes of Lagrangian methods which are not addressed in this work, namely various Mesh-Free methods and in particular the widely used Smooth Particle Hydrodynamics (SPH) method, initially introduced in the field of astro-physics [17]. SPH is a class of methods that represents the fluid continuum by a set of discrete particles. These particles have a distance (known as the "smoothing length"), over which their properties are "smoothed" by a kernel function. This means that the physical quantity of a particle can be obtained by summing up the relevant properties of all the particles which lie within the range of the kernel. The contributions of each particle to a property are weighted according to their distance from the particle of interest, and their density. Mathematically, this is governed by the kernel function, usually the Gaussian function and the cubic splines. SPH methods differ greatly from the FEM with respect to the choice of interpolation functions. For these methods the reader is referred to [18], [19].

1.3 Objectives and structure of the manuscript

The main objective of this work is to emphasize the advantages and disadvantages of Lagrangian FE-based methods in fluid mechanics, explore their limits and define strategies where the best of them can be taken. When deriving a new method it is often more important to clearly state its weaknesses and limitations, rather than just prove its applicability and highlight the advantages. In computational mechanics there exist no universal method, and by no means one can claim universal superiority of the proposed methodologies.

The emphasis on the solution of coupled problems is made. The approaches developed or presented in this work are viewed from the perspective of their benefits in the context of coupled or, in particular, fluid-structure interaction problems. Among the interaction set-ups we address three qualitatively different approaches in the course of this work: a partitioned fluid-structure interaction

strategy on moving meshes for fluid and solid (**Chapter 2**), a monolithic Lagrangian FSI approach (**Chapter 3**) and an Eulerian-Lagrangian immersed boundary-like approach (**Chapter 5**), that is used in this work for modeling fluid-gas interaction. **Chapter 5** also accounts for thermal interaction. This and the fact that we show that the Lagrangian formulation is beneficial for thermal problems with fluid undergoing properties alteration precludes us to restrict the title of this work to "fluid-structure interaction" and rather use a more general term "coupled problems".

The manuscript is organized as follows:

The first part (**Chapters 2** and **3**) deals with two versions of the Particle Finite Element Method (PFEM) and associated fluid-structure interaction solution strategies. **Chapter 2** presents the general PFEM philosophy as well as the classical PFEM-based incompressible fluid formulation. The main conceptual feature of the method which is the nodal variable storage is introduced.

Mesh re-generation by means of Delaunay triangulation/tetrahedralization and the alpha shape technique for boundary identification is presented. Additional ad-hoc techniques such as adaptive time-step and element quality checks that assure that no element becomes inverted are presented. The possibility of using ALE (Arbitrary Lagrangian Eulerian) fluid description instead of the Lagrangian one keeping the overall PFEM philosophy is mentioned.

In the second part of **Chapter 2** a partitioned strategy is developed for the solution of strongly coupled FSI involving a free-surface incompressible flow and a rigid body. The rigid body equations are presented. A modification of the Dirichlet-Neumann coupling by means of introducing the interface Laplacian (according to [20]) is presented. The chapter concludes with an industrial example dealing with the sea-landing of a satellite capsule.

Chapter 3 is devoted to a methodology for modeling the interaction between incompressible fluids and light-weight structures, an area in which most of standard partitioned FSI approaches fail. For this kind of applications an innovative updated Lagrangian quasi-incompressible fluid formulation is developed. It provides an alternative to the classical PFEM, described in the **Chapter 2**. The idea is to take the maximum advantage of using the Lagrangian formulation for the fluid in the sense of the similarity of the resulting equations to the ones of the solid. Lagrangian fluid equations can be written in a way very similar to the standard Lagrangian displacement-based structure, provided the fluid pressure variable is condensed in some way. This is facilitated by introducing a slight compressibility in the fluid. Once pressure is condensed, a monolithic FSI system can be established naturally by means of a standard FE assembly procedure, where the assembler and the solver do not differentiate between the fluid and the structure. The difference is only reflected in the entries in the unique global generalized stiffness matrix contributed either by the fluid or

the structure. From this point of view the quasi-incompressible formulation proposed here can be viewed as an extension of the ideas of Idelsohn et al [21], with a modification that allows using a much higher compressibility constants for the fluid. The methodology proposed here does not lead to the volumetric locking phenomenon, therefore assuring that the solution does not underestimate the velocity/displacement field for any given value of the compressibility constant. It is shown that for the practically optimal values of the bulk modulus the method conserves mass very well. The chapter presents in detail such issues as pressure condensation, pressure stability and efficient implementation. The non-linear and iterative linear solvers convergence details are presented. The chapter concludes with a number of examples showing the functionality of the method and its validation by means of comparisons with experiments and results found in the literature.

The second part of the work (**Chapters 4** and **5**) is devoted to numerical strategies where Eulerian and Lagrangian flow formulations are used together leading to an efficient approach for simulation of a complex thermo-mechanical phenomenon.

Chapter 4 presents an Eulerian compressible fluid formulation that shall be used in **Chapter 5**, where it shall be combined with the formerly developed Lagrangian formulations. The method is based upon the fully compressible Navier-Stokes equations equipped with the ideal gas equation of state. Only sub-sonic flow regimes are considered.

In our approach we propose a semi-explicit fractional-step-based solution scheme that on one hand is computationally efficient, and on the other hand does not have the limitations that a fully explicit scheme would have faced in terms of critical time step in the presence of acoustic waves at low Mach number. An efficient modular implementation is achieved by separating the energy equation from the momentum/continuity ones. This can be achieved by explicit time integration of the energy equation. The momentum/continuity system is split by means of the fractional step technique, finally dividing the monolithic system into a set of sub-systems. The resulting solver is validated using benchmark examples. Comparisons with the Boussinesq and low Mach number hypothesis based results are provided.

Chapter 5 deals with the coupling of Lagrangian and Eulerian fluids under the assumption that the mass ratio between the two fluids is small $\frac{\rho_E}{\rho_L} << 1$. We show, that using the Lagrangian formulation for a free surface flow inside the Eulerian ambient fluid effectively simplifies the problems connected with the moving interface identification and treatment that would have been faced if the Eulerian approach had been used for both fluids. On the other hand, the surrounding air that occupies the major portion of the computational domain, can be modeled more efficiently by an Eulerian approach. This way, re-meshing is performed only on the minor part of the domain

(polymer), thus not affecting significantly the computational cost.

The assumption of large density differences ($\rho_E << \rho_L$) justifies the simplified weak coupling approach. We present a methodology for the approximate imposition of the interface Dirichlet boundary conditions, that represent an effect of the motion of a "heavy" Lagrangian fluid onto the "light" ambient domain. The thermal problem (energy equation) is solved in a monolithic way, by projecting the thermal properties of the Lagrangian fluid onto the Eulerian mesh and the subsequent solution of the entire problem on the Eulerian mesh. Thus the thermal interaction is dealt with automatically. The chapter concludes with a practical example (of the modeling the polymer melting exposed to the heat flux due to fire) that motivated the development of the method. We emphasize that the methodology proposed in the chapter provides a framework for the solution of a class of thermally coupled problems that involve Lagrangian and Eulerian sub-domains. It could be enriched and further extended by inclusion of more physical phenomena, such as e.g. radiation and combustion.

Chapter 6 summarizes the entire work and gives a critical overview of the methods involving Lagrangian FE-based fluids. Innovative features of the work are highlighted. Conclusions are drawn with regard to the fields of application and the intrinsic limits of the presented methods. Future steps for the development of methods involving monolithic Lagrangian FSI and coupled Eulerian-Lagrangian algorithms for polymer melting simulation are outlined.

All the formulations and algorithms were implemented by the author within the Kratos Multi-Physics FE System, a C++ object oriented FE framework developed at CIMNE [22].

1.4 Notation

The following notation is generally utilized if not specified otherwise

By an overbar $\bar{(\cdot)}$ we denote vectors of nodal variables (in bold if the variable has several spatial components, e.g. $\bar{\mathbf{v}}$ is the nodal velocity vector, whereas $\bar{p}$ is the nodal pressure vector).

symbol	variable	unit
ρ	density	$[kg/m^3]$
μ	dynamic viscosity	$[Pa \cdot s]$
κ	bulk modulus	$[Pa]$
E	Young's modulus	$[Pa]$
ν	Poisson's ratio	$[Pa]$
c	speed of sound	$[m/s]$
t	time	$[s]$
Δt	time step	$[s]$
τ	stabilization parameter	$[s]$
$\mathbf{x}$	coordinate	$[m]$
$\bar{\mathbf{x}}$	vector of nodal coordinates	$[m]$
$\mathbf{x}_0$	initial position	$[m]$
$\bar{\mathbf{x}}_0$	vector of initial position of the nodes	$[m]$
$\mathbf{u} = \mathbf{x} - \mathbf{x}_0$	displacement	$[m]$
$\bar{\mathbf{u}}$=$\bar{\mathbf{x}} - \bar{\mathbf{x}}_0$	vector of nodal displacements	$[m]$
$\mathbf{v}$	velocity	$[m/s]$
$\bar{\mathbf{v}}$	vector of nodal velocities	$[m/s]$
$\mathbf{a}$	acceleration	$[m/s^2]$
$\bar{\mathbf{a}}$	vector of nodal accelerations	$[m/s^2]$
$\mathbf{f}$	body force	$[N]$
$\mathbf{F}$	vector of nodal body forces	$[N]$
V	volume	$[m^3]$
$\bar{V}$	vector of volumes assigned to the nodes	$[m^3]$
p	pressure	$[Pa]$
$\bar{p}$	vector of nodal pressures	$[Pa]$
$\mathbf{w}$	displacement test function	$[m]$
q	pressure test function	$[Pa]$
$\mathbf{r}$	dynamic residual	$[N]$
N	number of nodes	
δ_{kl}	Kronecker delta	
k,l	indices referring to the spatial components of a vector	
i	iteration index	
I,J	nodal indices	
n	time step index	

Table 1.1: Notation for the variables, properties and indices

symbol	discrete operator
$\mathbf{B}$	strain-displacement matrix
$\mathbf{K}$	stiffness matrix
$\mathbf{M}$	lumped displacement mass matrix
$\mathbf{M}_p$	lumped pressure mass matrix
$\mathbf{M}_p^c$	consistent pressure mass matrix
$\mathbf{D}$	divergence matrix
$\mathbf{G}$	gradient matrix
$\mathbf{L}$	Laplacian matrix
$\mathbf{H}$	dynamic tangent matrix
$\mathbf{C}_k$	volumetric elasticity tensor

Table 1.2: Relevant matrices

Chapter 2

The PFEM and partitioned FSI analysis

2.1 Introduction and outline

This chapter introduces the Particle Finite Element Method (PFEM) that shall serve as the fundamental framework for the formulations presented in this and the next chapter. First, the general concepts will be outlined and then the incompressible fluid formulation in the Lagrangian framework will be presented. This defines what is known as the classical PFEM [1], [2]. However this formulation will be presented in a descriptive way as it does not present the novelty of the current work. Issues relevant for an efficient implementation as well as some improvements of the original algorithms will be introduced.

Next a partitioned fluid-structure coupling strategy will be presented under the assumption that the structure is modeled as a rigid body while PFEM is used for modeling the fluid. Governing equations of the rigid body are introduced. The fluid-structure coupling based on a Dirichlet-Neumann scheme enriched by an "interface Laplacian" at the fluid-structure boundary [20] will be introduced. The derivations of this chapter were motivated by an industrial project, whose goal was the analysis of the impact of a satellite capsule against the sea-surface. This shall serve as an example showing the functionality of the proposed FSI methodology at the end of the chapter.

2.2 The Particle Finite Element Method

The PFEM adopts an updated Lagrangian framework for the description of the fluid, where the mesh nodes are treated as particles that can freely move and even separate from the main fluid domain. The fundamental idea of the PFEM is that the variables of interest are stored at the nodes instead of the Gauss points[1]. This results in a hybrid between a standard FE and a mesh-free method. A finite element mesh is created at every time step of the dynamic problem and the solution is then stored at the nodes. The nodes move according to their velocity obtaining their new position and then the mesh is re-created. The generation of a FE mesh is done using a Delaunay triangulation [23]. In our approach we use simplicial triangular/tetrahedral meshes. In treating problems involving free surface flows the boundary is determined at every time step using the so-called 'alpha-shape'[2] technique [24], [1].

It is important to keep in mind that the convective term of the momentum equation disappears in the Lagrangian description. Therefore the problem remains elliptic and the discrete system is symmetric. Thus the stability problems faced in Eulerian methods due to the presence of the convective term do not exist in Lagrangian approaches.

Incompressible fluid governing equations in the Lagrangian framework An incompressible fluid in the Lagrangian framework is governed by Navier-Stokes equations that can be written for a viscous Newtonian fluid:

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \nabla p - \nabla \cdot (2\mu\epsilon(\mathbf{v})) = \rho\mathbf{g} \tag{2.1}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{2.2}$$

where $\mathbf{v}$ is the velocity vector, p the pressure, t - time, $\mathbf{g}$ the body force, ρ the density, dynamic viscosity μ and $\epsilon = \frac{\nabla\mathbf{v}+\nabla^T\mathbf{v}}{2}$ - the deviatoric strain rate. Even though the momentum equations do not need to be stabilized in the Lagrangian framework, the continuity equation requires stabilization for equal order velocity-pressure interpolations [25]. The details regarding pressure instabilities origin are given in the next chapter. Here we illustrate the pressure stabilization using the Finite Calculus method (FIC) [26].

[1] Storing the variables at the nodes permits the use of re-meshing without information loss and necessity of mapping between the old and new meshes, that would have been necessary in case of standard Gauss points storage. Possibility of efficient re-meshing is essential for a Lagrangian fluid formulation where severe domain deformations are expected.

[2] Alpha shape is a technique that permits one to reconstruct a shape from a dense arbitrary set of data points.

Let us define first the residual of the momentum and continuity equations as

$$\mathbf{r}_m = \rho\frac{\partial \mathbf{v}}{\partial t} + \nabla p - \nabla \cdot (2\mu\epsilon(\mathbf{v})) - \rho\mathbf{g} \tag{2.3}$$

$$r_c = \nabla \cdot \mathbf{v} = \frac{\partial v_i}{\partial x_i} \tag{2.4}$$

one can write the momentum and continuity equation as

$$\mathbf{r}_m = 0 \tag{2.5}$$

and

$$r_c = 0 \tag{2.6}$$

The stabilized continuity equation according to the FIC approach is obtained as follows [1]:

$$r_c - \frac{1}{2}h_j\frac{\partial r_c}{\partial x_j} = 0 \tag{2.7}$$

where h_j is the characteristic length of the domain in the direction x_j, where the continuity equation is enforced. Details on derivation of the modified continuity equation can be found in [26], [27].

The stabilization term in Eq. (2.7) can be expressed in terms of the momentum residual [28], thus giving the following form of the stabilized continuity equation [1]

$$r_c - \sum_i \tau_i \frac{\partial r_{m_i}}{\partial x_i} = 0 \tag{2.8}$$

with τ_i being the algorithmic stabilization parameter with the dimension of time over density, defined as

$$\tau_i = \frac{3h_i^2}{8\mu} \tag{2.9}$$

Weak form and discretization The weak form of the problem defined by Eqs. (2.5) and (2.8) reads:

$$(\mathbf{w}, \mathbf{r}_m) = 0 \tag{2.10}$$

and

$$\left(q, r_c - \sum_i \tau_i \frac{\partial r_{m_i}}{\partial x_i}\right) = 0 \tag{2.11}$$

where $\mathbf{w}$ and q are velocity and pressure test functions and $(a, x) := \int_\Omega axd\Omega$ is a bilinear form.

The computation of Eq. (2.11) is facilitated if a new variable, called the "pressure gradient projection" π is introduced as:

$$\pi_i = r_{m_i} - \frac{\partial p}{\partial x_i} \tag{2.12}$$

We express the momentum residual in terms of this newly-introduced variable and after integration by parts we end up with the following system of governing equations

$$(\mathbf{w}, \mathbf{r}_m) = 0 \tag{2.13}$$

$$(q, r_c) + \int_\Omega \sum_i \tau_i \frac{\partial q}{\partial x_i} \left(\frac{\partial p}{\partial x_i} + \pi_i \right) d\Omega = 0 \tag{2.14}$$

$$\left(\tau_i \psi, \frac{\partial p}{\partial x_i} + \pi_i \right) = 0 \tag{2.15}$$

where ψ is the pressure gradient projection test function. Note that the boundary term resulting from the integration by parts of $\mathbf{r}_m$ in Eq. (2.14) has been neglected as the influence of this term in the numerical solution has been found to be negligible [1]. The semi-discrete form of the above equations using a standard linear velocity-pressure interpolation ($(\bar{\mathbf{v}}, \bar{p}, \bar{\pi}) = \sum_{I=1}^{n} N_I \left(\bar{\mathbf{v}}_I, \bar{p}_I, \bar{\pi}_I \right)$, where I is the nodal index, $(\cdot)_I$ denotes a nodal variable, N are the linear shape functions) reads (note that the equilibrium is written in the current configuration $\mathbf{X}_t$, adopting updated Lagrangian approach):

$$\mathbf{M} \frac{\partial \bar{\mathbf{v}}}{\partial t} + \nu \mathbf{L} \bar{\mathbf{v}} + \mathbf{G} \bar{p} = \mathbf{F} \tag{2.16}$$

$$\mathbf{D} \bar{\mathbf{v}} + \mathbf{L}_\tau \bar{p} + \mathbf{Q} \bar{\boldsymbol{\pi}} = 0 \tag{2.17}$$

$$\mathbf{Q}^T \bar{p} + \hat{\mathbf{M}} \bar{\boldsymbol{\pi}} = 0 \tag{2.18}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{L}$ the Laplacian matrix, $\mathbf{G}$ the gradient matrix, $\mathbf{Q}$ the stabilization gradient matrix and $\mathbf{D}$ the divergence matrix at time t. Note that by $\bar{(\cdot)}$ we denote vectors of nodal variables (in bold if the variable has several spatial components, e.g. velocity $\bar{\mathbf{v}}$ and pressure gradient projection $\bar{\boldsymbol{\pi}}$).

These matrices are assembled from the element contributions defined as

$$M_{IJlk} = \int_{\Omega_e} \rho \delta_{kl} N_I N_J d\Omega \tag{2.19}$$

$$L_{IJ} = \int_{\Omega_e} \frac{\partial N_I}{\partial x_k} \frac{\partial N_J}{\partial x_k} d\Omega \tag{2.20}$$

$$L_{IJ\tau} = \int_{\Omega_e} \tau \frac{\partial N_I}{\partial x_k} \frac{\partial N_J}{\partial x_k} d\Omega \tag{2.21}$$

$$G_{IJk} = \int_{\Omega_e} \frac{\partial N_I}{\partial x_k} N_J d\Omega \tag{2.22}$$

$$\mathbf{D} = \mathbf{G}^T \tag{2.23}$$

$$F_{Ik} = \int_{\Omega_e} N_I \mathbf{f}_k d\Omega \tag{2.24}$$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}^1 & \mathbf{Q}^2 \end{pmatrix} \quad Q^k_{IJ} = \int_{\Omega_e} \tau_k \frac{\partial N_I}{\partial x_k} N_J d\Omega \tag{2.25}$$

$$\hat{\mathbf{M}} = \begin{pmatrix} \hat{\mathbf{M}}^1 & 0 \\ 0 & \hat{\mathbf{M}}^2 \end{pmatrix} \quad \hat{\mathbf{M}}^k_{IJ} = \int_{\Omega} \tau_k N_I N_J \tag{2.26}$$

where Ω_e is the element integration domain at t.

Time integration using the generalized trapezoidal rule gives:

$$\mathbf{M} \frac{\bar{\mathbf{v}}_{n+1} - \bar{\mathbf{v}}_n}{\Delta t} + \nu \mathbf{L} \bar{\mathbf{v}}_{n+\theta} + \mathbf{G} \bar{p}_{n+1} = \mathbf{F}_{n+\theta} \tag{2.27}$$

$$\mathbf{D} \bar{\mathbf{v}}_{n+1} + \mathbf{L}_\tau \bar{p}_{n+1} + \mathbf{Q} \bar{\boldsymbol{\pi}}_{n+1} = 0 \tag{2.28}$$

$$\mathbf{Q}^T \bar{p}_{n+1} + \hat{\mathbf{M}} \bar{\boldsymbol{\pi}}_{n+1} = 0 \tag{2.29}$$

where $\bar{\mathbf{v}}_{n+\theta} = (1-\theta)\bar{\mathbf{v}}_n + \theta \bar{\mathbf{v}}_{n+1}$ with $0 \leq \theta \leq 1$. Note that it is usual to use $\bar{p}_{n+1}$ instead of $\bar{p}_{n+\theta}$, as it is sufficient to ensure that the divergence free-condition for the velocity is satisfied at any instance between t_n and t_{n+1}. This option is followed e.g. in [29].

It is convenient to condense the third equation $\bar{\boldsymbol{\pi}} = -\hat{\mathbf{M}}^{-1} \mathbf{Q}^T \bar{p}$, substituting the pressure gradient projection into the continuity equation thus yielding the following system:

$$\mathbf{M} \frac{\bar{\mathbf{v}}_{n+1} - \bar{\mathbf{v}}_n}{\Delta t} + \nu \mathbf{L} \bar{\mathbf{v}}_{n+\theta} + \mathbf{G} \bar{p}_{n+1} = \mathbf{F}_{n+\theta} \tag{2.30}$$

$$\mathbf{D} \bar{\mathbf{v}}_{n+1} + \mathbf{L}_\tau \bar{p}_{n+1} + \mathbf{Q} \hat{\mathbf{M}}^{-1} \mathbf{Q}^T \bar{p}_{n+1} = 0 \tag{2.31}$$

Finally for the sake of convenience we introduce the notation $\mathbf{S} = \mathbf{Q} \hat{\mathbf{M}}^{-1} \mathbf{Q}^T + \mathbf{L}_\tau$, permitting us to

write the final system as

$$\mathbf{M}\frac{\bar{\mathbf{v}}_{n+1}-\bar{\mathbf{v}}_n}{\Delta t}+\nu\mathbf{L}\mathbf{v}_{n+\theta}+\mathbf{G}\bar{p}_{n+1}=\mathbf{F}^{n+\theta} \tag{2.32}$$

$$\mathbf{D}\bar{\mathbf{v}}_{n+1}+\mathbf{S}\bar{p}_{n+1}=0 \tag{2.33}$$

The system of Eqs. (2.32-2.33) can either be solved monolithically (i.e. for pressure and velocity variables at once), or by means of a fractional step scheme [30], [31], [29], where velocity and pressure solution steps are decoupled.[3] The fractional step method is based on the solution of the momentum equations for an intermediate (non-solenoidal) velocity $\tilde{\mathbf{v}}$ and a subsequent correction performed to obtain the end-of-step velocity $\mathbf{v}^{n+1}$. We shall present the derivation of the fractional step equations in some detail, as it constitutes an important technique relevant also for the following chapters.

The set of fractional momentum equations can be written as

$$\mathbf{M}\frac{\tilde{\mathbf{v}}-\bar{\mathbf{v}}_n}{\Delta t}+\nu\mathbf{L}\tilde{\mathbf{v}}+\gamma\mathbf{G}\bar{p}_n=\mathbf{F}_{n+\theta} \tag{2.35}$$

$$\mathbf{M}\frac{\bar{\mathbf{v}}_{n+1}-\tilde{\mathbf{v}}}{\Delta t}+\mathbf{G}\left(\bar{p}_{n+1}-\gamma\bar{p}_n\right)=0 \tag{2.36}$$

Adding these two equations together we recover the initial equation except for the viscous term. Thus the approximation of the fractional step in the Lagrangian framework originates from the approximation $\mathbf{L}\mathbf{v}_{n+\theta}\approx\mathbf{L}\tilde{\mathbf{v}}$ and does not influence the accuracy of the approach especially for low viscosity values. In Eulerian formulations the situation is different, because there the approximation includes also the convective term [29].

Parameter γ can be either chosen to be 0 or 1, resulting in 1st or 2nd order fractional step schemes. For detailed analysis and comparison of the two the reader is referred to [29]. The first order scheme consists in the solution of the fractional momentum equation that does not contain the volumetric term. The second order method "predicts" the volumetric term by using the end-of-previous step pressure. In this work the second order scheme is implemented due to its higher accuracy.

Next we apply the continuity equation Eq. (2.33) to the end-of-step momentum Eq. (2.36), thus

[3]The fractional step method is based on Hodge theorem stating that an arbitrary vector field can be presented as a sum of a divergence free vector field and a gradient of a scalar quantity. In the context of fluid dynamics the vector field is the velocity and the scalar quantity is the pressure. That is, an arbitrary velocity $\tilde{\mathbf{v}}$ can be represented as

$$\tilde{\mathbf{v}}=\mathbf{v}+\nabla p \tag{2.34}$$

where $\nabla\cdot\mathbf{v}=0$.

resulting in

$$\Delta t \mathbf{D}\mathbf{M}^{-1}\mathbf{G}\left(\bar{p}_{n+1}-\bar{p}_n\right)+\mathbf{S}\bar{p}_{n+1}=\mathbf{D}\tilde{\mathbf{v}} \tag{2.37}$$

Since construction of the $\mathbf{D}\mathbf{M}^{-1}\mathbf{G}$ is computationally expensive even if the mass matrix is taken in the lumped format, usually it is approximated by the operator $\mathbf{L}$, representing an approximation of the Laplacian operator, i.e.

$$\mathbf{D}\mathbf{M}^{-1}\mathbf{G}\approx\mathbf{L} \tag{2.38}$$

This gives us the final equations system characteristic for the fractional step method applied to the viscous incompressible fluid in the Lagrangian framework as:

$$\mathbf{M}\frac{\tilde{\mathbf{v}}-\bar{\mathbf{v}}_n}{\Delta t}+\nu\mathbf{L}\tilde{\mathbf{v}}+\mathbf{G}\bar{p}_n=\mathbf{F}_{n+\theta} \tag{2.39}$$

$$\Delta t\mathbf{L}\left(\bar{p}_{n+1}-\bar{p}_n\right)+\mathbf{S}\bar{p}_{n+1}=\mathbf{D}\tilde{\mathbf{v}} \tag{2.40}$$

$$\mathbf{M}\frac{\bar{\mathbf{v}}_{n+1}-\tilde{\mathbf{v}}}{\Delta t}+\mathbf{G}\left(\bar{p}_{n+1}-\bar{p}_n\right)=0 \tag{2.41}$$

As we see the velocity and pressure solution steps become decoupled. First, Eq. (2.39) is solved for $\tilde{\mathbf{v}}$ knowing $\bar{p}_n$ and $\bar{\mathbf{v}}_n$, then the end-of-step pressure $\bar{p}_{n+1}$ is computed from $\tilde{\mathbf{v}}$ (Eq. (2.40)). Finally, the end-of-step velocity is found from $\bar{p}_{n+1}$ and $\tilde{\mathbf{v}}$ according to Eq. (2.41).

Having developed the governing equations, we present next the second fundamental ingredient of the PFEM, namely the re-meshing step.

Re-meshing using Delaunay triangulation/tetrahedralization In the PFEM the mesh is re-generated at every time step of a dynamic problem using the Delaunay triangulation/tetrahedralization. When using the term "mesh re-generation" we slightly abuse the terminology since when the Delaunay technique is applied only the nodal connectivities (and thus elements) change , while the nodes themselves remain generally preserved. The only case in which the nodes are locally removed/added appears when adaptive re-meshing is carried out. Here we shall assume that the mesh nodes are preserved in the entire domain.

Delaunay triangulation for a set P of points in the plane is a triangulation $DT(P)$ such that no point in P lies inside the circumcircle of any triangle in $DT(P)$ [23], [32], [33] (see an example in Fig. 2.1). Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation; they tend to avoid deformed degraded triangles [23].

The circumcircle of a triangle formed by three points of P is "empty" if it does not contain vertices other than the ones that define it (other points are permitted only on the perimeter, not

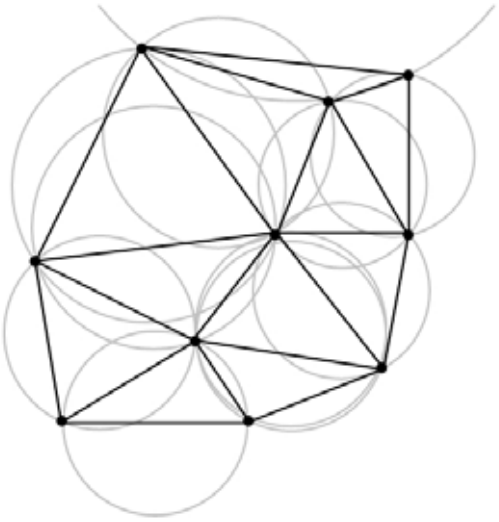


Figure 2.1: Delaunay triangulation of a point set

inside). The triangulation is "a Delaunay triangulation" if the circumcircles of all the triangles are empty. Similarly for 3D the definition is based on circumspheres.

For points lying on one straight line the triangulation is undefined. For four points on the same circle (e.g., the vertices of a rectangle) the Delaunay triangulation is not unique: clearly, the two possible triangulations that split the quadrangle into two triangles satisfy the Delaunay condition. This problem can be overcome by a particular algorithm.

The computing time required for the re-meshing using Delaunay triangulation is of the order $O\left(n \cdot log\, n\right)$ with n being the number of nodes. Actually it is possible to reduce the computational time associated to the Delaunay triangulation to nearly $O(n)$ by using spatial optimization schemes [33]. A FE-code that involves a re-meshing algorithm is numerically feasible as long as the associated computational effort is less than $O\left(n^2\right)$. Otherwise the computational effort associated to re-meshing would dominate the effort associated to the solution of the equation system.

Boundary identification As mentioned already, Lagrangian methods are especially beneficial in application to free-surface flows simulations. For these problems the correct definition of the domain boundary becomes necessary at every time step of a transient problem. We address this issue next.

Classical alpha-shape method in PFEM The Delaunay partition facilitates the recognition of the boundary nodes. Considering that the nodes follow a variable $h(x)$ distribution, where $h(x)$ is the minimum distance between two nodes, the following criterion defines the classical PFEM approach [1]: all nodes on an empty sphere with a radius greater than $\alpha\, h$, are considered as boundary nodes.

In practice α is a parameter close to, but greater than one (typically 1.3 in 2D and 1.5 in 3D). This criterion, is coincident with the "alpha-shape" concept of Edelsbrunner [32], and will be referred as "alpha-shape method" from now onwards.

As soon as the boundary nodes are identified, the triangles whose nodes (all nodes) belong to the boundary are erased. Then the boundary curve/surface is defined by all the polygons (or polyhedrons in 3D) having all their nodes on the boundary and belonging to just one polyhedron. Fig. 2.2 depicts the boundary identification using the alpha shape technique.

A combination of alpha-shape and boundary markers In the present work an improvement of the standard alpha-shape technique is proposed. It is possible to take advantage of the information regarding the boundaries at the step preceding re-meshing. A flag variable "boundary" was used to identify if a node belongs to a domain boundary ("free surface" or "wall"). In particular only elements containing at least one node with a non-zero "free surface" flag are passed to the alpha-shape check: if the radius r of the circle/sphere around the element is larger than $\alpha\, h$, then the element is erased and all its nodes are marked with a "boundary" flag set to "free surface". On the contrary, if the element is found to be an internal element (containing no "free surface" nodes), in our algorithm we remove it only if it is extremely distorted, i.e. using $\alpha_{in} = 10\,\alpha$ (index "in" stands for "internal" element). This is done in order to prevent creation of possible non-physical voids in the domain interior. The element is removed only if it is found to be so severely distorted to perturb the calculations. We preclude creation of the elements with all the nodes being "wall" nodes, independent of its geometrical properties.

The correct boundary curve/surface may be important to define the correct external normal to the curve/surface. Furthermore, in weak forms (Galerkin) a correct evaluation of the domain volume is also important. Nevertheless, it must be noted that in the criterion proposed above, the error in the boundary surface definition is proportional to h [1]. This is the error order accepted in a numerical method for a given node distribution. The only way to obtain a more accurate boundary surface definition is by decreasing the distance between particles.

The method described above also allows one to identify isolated fluid particles outside the main fluid domain. These particles are treated as part of the external boundary where pressure is fixed to the atmospheric value and their motion is described as a parabolic motion of a mass-point. When the particle joins the main fluid domain, it again becomes a FE node, and follows the continuum mechanics equations.

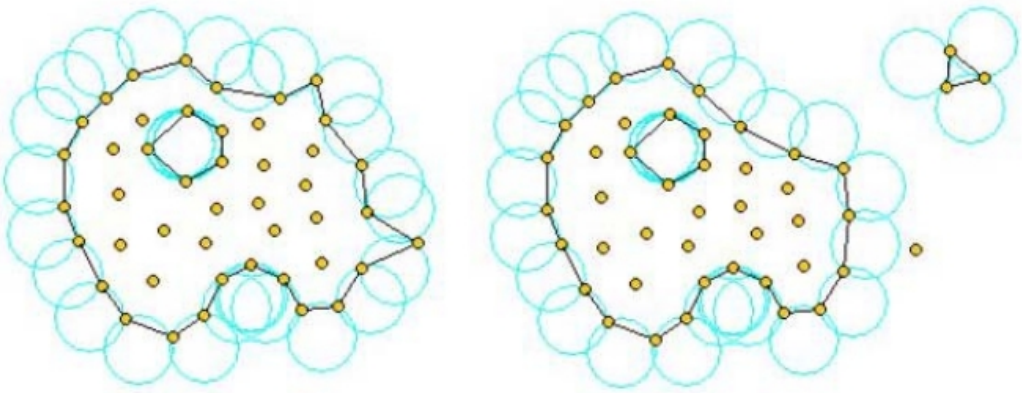


Figure 2.2: Alpha shape determination of the boundary [1]

Mesh quality control In Lagrangian formulations large mesh deformations are tackled by application of frequent re-meshing. [4] Delaunay triangulation guarantees that for a given node distribution the optimal element mesh is created, however the nodal distribution itself is completely determined by the solution of the governing continuum mechanics equations, thus precluding any external control (e.g. mesh smoothing) over the position of the nodes. Therefore, even if re-meshing is performed, there is no guarantee that the element inversion (which leads to an immediate failure of the solver) takes place. To overcome this problem, the time step size must be estimated in order to ensure that no element becomes inverted. [5]

Time step estimation Element degradation implies that its area may become zero or negative within one time step. Noting that the element Jacobian is equal to twice the elemental area ($J = 2A_{el}$), one can relate the element degradation to having a zero Jacobian and thus compute the critical time step. The degradation condition then reads

$$J = det\left(\frac{\partial \bar{\mathbf{x}}_{n+1}}{\partial \bar{\mathbf{x}}_n}\right) = 0 \tag{2.42}$$

Knowing the position and velocity of the nodes of an element at time step n one can estimate the position of these nodes at time step $n+1$. It is sufficient to perform a first order prediction,

[4] in practice at every time step

[5] Note that this time step estimation is different from the one performed for the stability assurance of conditionally-stable time integration schemes and is rather a geometrical criterion that needs to be carried out irrespectively of the stability time estimate

that is: $\bar{\mathbf{x}}_{n+1} \approx \bar{\mathbf{x}}_n + \bar{\mathbf{v}}_n \Delta t$. Then, expanding 2.42 we arrive at

$$\mathbf{J} = det\left(\frac{\partial \bar{\mathbf{x}}_{n+1}}{\partial \bar{\mathbf{x}}_n}\right) = det\left(\frac{\partial \bar{\mathbf{x}}_n}{\partial \bar{\mathbf{x}}_n} + \Delta t \frac{\partial \bar{\mathbf{v}}_n}{\partial \bar{\mathbf{x}}_n}\right) = det\left(\mathbf{I} + \Delta t \frac{\partial \bar{\mathbf{v}}_n}{\partial \bar{\mathbf{x}}_n}\right) \tag{2.43}$$

where $\mathbf{I}$ is the identity matrix. To find the critical time step, one has to solve the equation: $det\left(\mathbf{I} + \Delta t_{crit} \frac{\partial \bar{\mathbf{v}}_n}{\partial \bar{\mathbf{x}}_n}\right) = 0$ for Δt_{crit} that would correspond to the element degradation[6]. It is meaningful to introduce a safety factor s and to use e.g. $\Delta t = s \cdot \Delta t_{crit}$ with s usually chosen to be equal to 0.5. Application of adaptive time stepping assures that no element is degraded or inverted during the solution.

Adaptive re-meshing: nodal removal The time step estimation approach ensures that no element is inverted, however the estimated time step can become extremely small in presence of distorted elements. To be able to work with reasonable time steps, mesh quality should be considered. If two nodes get close to each other, one of them must be eliminated. However the overall mesh resolution should not suffer from this procedure. This is obtained by introducing a criterion for removing the nodes and imposing an element size constraint for keeping the mesh size within desirable limits, which includes local refinement and de-refinement. The criterion for removal of nodes is based upon the ratio between the element edge length $a1$ and the length of the remaining edges (e.g. a_2 and a_3 in 2D) reflecting the degree of distortion. Another possibility is comparing all the edge lengths a_i with the desired edge length a_d that should be prescribed. We considered reasonable to remove nodes if this ratio exceeds the value of $\approx 30\%$. This criterion is somewhat arbitrary and can be replaced by any other user-defined value.

Another problem might appear even when all edges have an admissible length, that is when a node approaches the wall or in general its counter-edge. Obviously in this situation even though no nodes are close to each other, the element is approaching degradation. In order to identify this situation one uses the described above alpha shape technique, in order to compare the height of the element with the radius of the circle/sphere around its nodes. If the ratio $\frac{h}{r}\,\alpha < 1$ the marked node shall be removed. Both cases are depicted in the Fig. 2.3. These techniques ensure that nodes do not get too close to each other or to a reciprocal edge/facet.

Adapting re-meshing: keeping the mesh resolution As we already mentioned, the nodal removal should not impoverish the mesh resolution. In order to keep the mesh quality or if necessary even refine it (e.g. based on error estimation or some other criterion), an adaptive meshing interface

[6] One can also use a second order prediction for $\bar{\mathbf{x}}_{n+1}$ by considering also the accelerations

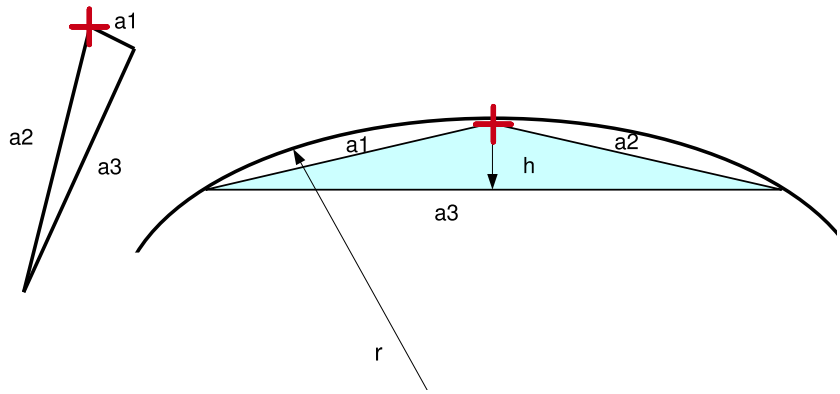


Figure 2.3: Geometrical criterion for nodal removal in order to ensure the mesh quality

was implemented using the "Triangle" [34] (in 2D) and "Tetgen" [35] (in 3D) meshers. We briefly explain its functionality: every node stores the so-called initial "nodal length" h_i, which is average inter-nodal distance between the node i and the nodes of the patch around it in the undeformed initial mesh. The volume of a perfect (equidistant) tetrahedron is calculated as $\frac{1}{12}(\frac{3}{2}h)^3\sqrt{2}$, where h is the average of its "nodal" lengths, i.e. $h_{el} = \frac{1}{4}(h_1 + h_2 + h_3 + h_4)$ for a 4-noded tetrahedra. Given the nodal h distribution, the mesher is asked to create elements with the volume smaller or equal to the volume of the perfect tetrahedron with edge size h and if necessary introduces new nodes to complete the criterion. In fact, instead of trying to keep the initial size distribution, it is possible to provide new nodal h maps in an ad-hoc manner during the runtime.

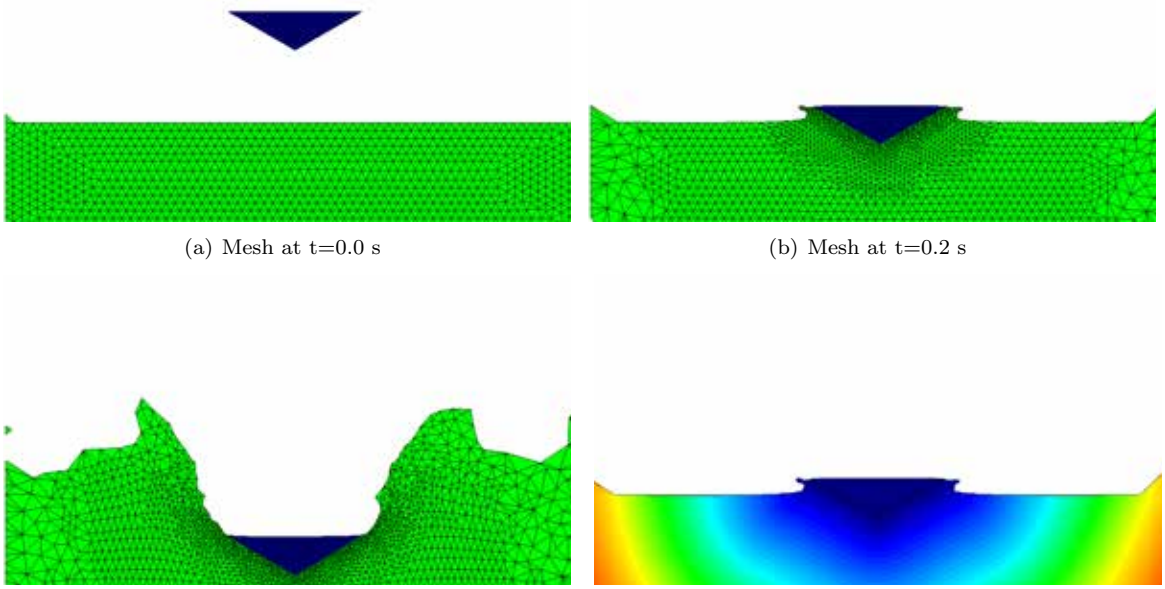
This is of particular importance in the fluid-structure interaction (FSI) simulations. There, one needs a fine mesh resolution close to the object, while mesh can be gradually de-refined as the distance grows. It was found efficient to prescribe the mesh size based on the function proportional to the square of the distance between the object and a fluid node (distance is calculated as the minimal distance between the current fluid node and a node of solid)[7]. If we call this distance x, the maximum and minimal element height h_{max} and h_{min} respectively, and the maximum distance from the object to the fluid nodes d_{max}, the formula for the element height h reads:

$$h = \frac{h_{max} - h_{min}}{d_{max}^2} x^2 + h_{min} \tag{2.44}$$

An example of the adaptive refinement process is illustrated in the Fig.2.4. An elastic wedge discretized with a fine mesh falls into a water container. A rough mesh is initially used for the water discretization. Our interest is to have fine mesh resolution in the water domain close to the object

[7]Instead of this "geometrical" adaptivity criteria one can use an error estimator (developed for a Lagrangian fluid in [36], [37]).

and keep the mesh coarse or even de-refine it as the distance grows. The refinement is based upon the distance from the object. The contours of the distance function are shown in the Fig. 2.4 (d):



(a) Mesh at t=0.0 s

(b) Mesh at t=0.2 s

(c) Mesh at t=0.3 s

(d) Distance function at t=0.2 s

Figure 2.4: Mesh refinement: an example of the sinking wedge

Leakage Adaptive time stepping ensures absence of inverted elements and adaptive re-meshing preserves the mesh quality, however there exist a danger of "leakage". This is illustrated in Fig. 2.5: One can see a node with velocity orthogonal to the wall it is approaching (shown in yellow), where the zero-velocity b.c. is applied. If the node comes close enough to the wall, the mesher removes the corresponding element as it becomes distorted. At the subsequent time step the node can move freely and penetrates through the wall. As the edge of the wall element does not exist anymore[8], this "hole" remains during the rest of the simulation and the fluid leaks out. In order to avoid leakage one has to remove the nodes moving towards the fixed domain boundaries as soon as they approach closer than a critical distance. This distance is typically taken as equal to half of the average inter-nodal distance[9].

[8] This defines one of the drawbacks of using the unconstrained Delaunay, where the mesher receives exclusively information associated to the nodes, ignoring the information associated e.g. to the edges.

[9] Additional mesh quality control can be achieved by formulating the PFEM in the Arbitrary Lagrangian Eulerian (ALE) framework. In the ALE approach the movement of the fluid particles is decoupled from that of the mesh nodes, but in contrast with Eulerian description, the mesh is allowed to move in an "arbitrary" way. Hence the relative velocity between mesh nodes and particles is used as the convective velocity in the momentum equations.

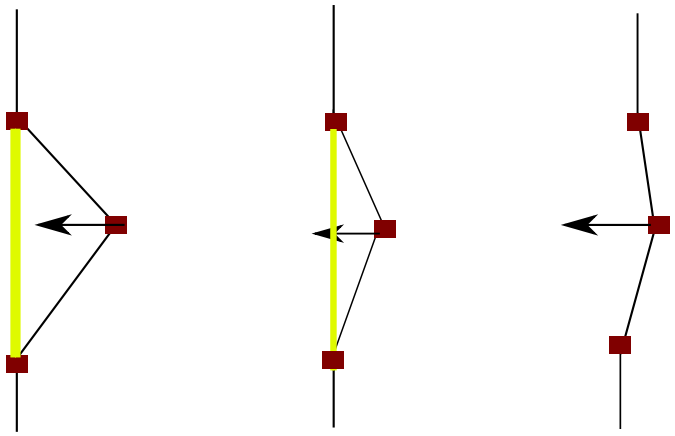


Figure 2.5: Node passing through wall

Algorithm At this point all the essential ingredients of the PFEM are presented. Table 2.1 summarizes the application of PFEM to the solution of a hydrodynamic problem:

1. Discretize the fluid with a finite element mesh.
2. Identify the external boundaries for the fluid (using improved alpha-shape technique)
3. Solve the Lagrangian equations of motion for the fluid (fractional step technique in our case). Compute the relevant state variables: velocities, pressure and displacements
4. Move the mesh nodes to a new position (according to the computed incremental displacement).
5. Re-generate mesh for the fluid domain
6. Go back to the next time step. Start solution from Step 2

Table 2.1: Algorithm for solution of a hydrodynamic problem using PFEM

The Navier-Stokes equations for the incompressible fluid then take the following form

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} - \mathbf{v}_M)\cdot\nabla\mathbf{v} + \nabla p - \mu\epsilon(\mathbf{v}) = \mathbf{f} \tag{2.45}$$

$$\nabla\cdot\mathbf{v} = 0 \tag{2.46}$$

The ALE approach inherits the advantages of the Lagrangian method in terms of necessity of discretization of the real physical domain only and in the natural definition of the fluid-structure interface. However since the convective term appears in Eq. (2.45) stabilization techniques for the convection need to be introduced as well as an efficient mesh-moving algorithm. One option for the mesh-moving algorithm consists in first performing the Lagrangian motion (as in classical PFEM), and then performing the mesh adjustment, smoothing the position of the nodes (except the ones that are defining the boundaries, e.g. the free surface) using conventional techniques, such as Gaussian or Laplacian smoothing. Mesh smoothing leads in principle to an improved distribution of the nodes and therefore to better quality of the elements. Additionally the "leakage" phenomenon can be avoided without removing the nodes approaching the wall. Instead it is possible simply to remove the mesh velocity component orthogonal to the wall of a wall-approaching-node. This can be achieved exclusively due to the decoupling of the mesh and material velocity.

2.3 Partitioned FSI

Partitioned methods define the class of methods used most widely in modeling fluid-structure interaction (FSI) problems [38], [39]. They rely on an independent solution of the sub-systems involved and enable re-usability of the existing codes for each sub-system (the fluid and the structure). Their counterpart are the monolithic approaches, based on the solution of single discrete equation system, describing the whole system. In either of the approaches the equations to be solved separately (partitioned) or together (monolithic) are the Navier-Stokes equations for the fluid, momentum equation for the solid and usually the interface equation. The interface equation must enforce the dynamic and kinematic coupling conditions. That can be written as

$$\bar{\mathbf{v}}_\Gamma = \frac{d\bar{\mathbf{u}}_\Gamma}{dt} \tag{2.47}$$

$$\bar{\boldsymbol{\sigma}}_\Gamma^S \cdot \mathbf{n} = \bar{\boldsymbol{\sigma}}_\Gamma^F \cdot \mathbf{n} \tag{2.48}$$

where $\bar{\mathbf{v}}_\Gamma$ and $\bar{\mathbf{u}}_\Gamma$ are the velocity of the fluid and displacement of the structure at the interface; $\boldsymbol{\sigma}$ is the stress tensor (superscripts stand for "solid" and "fluid") and $\mathbf{n}$ is the interface normal.

Monolithic approaches are generally more expensive as they require the solution of a single large equation system instead of several smaller sub-systems as in the partitioned approaches. They are, nevertheless, known to be more robust, as they do not introduce any domain-splitting error. The main disadvantage of monolithic methods is the poor conditioning of the resulting discrete system due to the different physical nature of the constituents, reflected in different scaling of variables describing the sub-domains [38]. We leave the detailed description of the monolithic methods for the next chapter and here will concentrate our attention on the partitioned approaches.

Kinematic description of the sub-domains in FSI problems When a partitioned approach is chosen, one has to define the way in which the sub-domains communicate among themselves. This is strongly related with the kinematic description chosen for each sub-domain.

One typical option relies on the Eulerian description for the fluid and the Lagrangian one for the structure. From the point of view of code modularity and re-usability this option is highly advantageous as the standard fluid codes are usually based on the Eulerian description while structural codes use the Lagrangian framework. Thus, for the FSI such an approach would theoretically require definition and implementation of the coupling only, that models the "communication" of the individual codes over the interaction interface. Such approaches are classified sometimes as "immersed

boundary" methods [40]. In this setting the Lagrangian mesh of the structure is moving on top of the fixed fluid mesh and forces the fluid material corresponding to the image of the Lagrangian structure on the fluid mesh to deform as the structure. The interface tracking requires computation of the intersections of the Lagrangian mesh with the Eulerian one at every solution step. Also the interaction boundary conditions prescription becomes involving since the Eulerian fluid description disables the possibility matching the deforming Lagrangian mesh at the interface. In any case the coupling in the context of this approach requires additional techniques including numerous mesh-to-mesh mappings, data transfer and spatial searches. We explore the Eulerian-Lagrangian setting in **Chapter 5**.

The Eulerian fluid-Eulerian solid option is not beneficial in the partitioned context. With respect to existing codes re-utilization, this approach does not provide any advantage, since the vast majority of existing structural codes use the Lagrangian setting. Such strategy would require re-definition of the structural code in Eulerian frame, which is cumbersome. Additionally, while the problem of interface identification in Eulerian-Lagrangian methods consisted only in mapping the position of the solid (explicitly defined by its Lagrangian nature) onto the Eulerian mesh, in the Eulerian-Eulerian setting the situation becomes more complicated since even determining the position of the moving solid domain would require additional techniques, such as convection of the distance function. However the Eulerian-Eulerian approach can lead to significant advantages in a monolithic context, since adopting same kinematic framework for the fluid and the solid leads to a similar form of the discrete governing equations, facilitating their simultaneous solution [41].

A third option consists in applying the mesh-moving approaches for both the sub-domains and is followed here. Using the fully Lagrangian or the ALE PFEM in the context of fluid-structure interaction does not introduce any additional difficulty in comparison to single-field fluid problems. The interfaces are defined naturally and the use of matching interface meshes facilitates application of the boundary conditions. The re-meshing needs to be performed exclusively on the fluid domain. The fluid and the structure start interaction as soon as a fluid node approaches the mesh of structure within a critical distance h_c. Then the mesher creates the fluid element connecting both sub-domains. This is shown in Fig. 2.6. At time step t_n the distance between the fluid and structure meshes is larger than h_c and thus no contact takes place. In the next step t_{n+1} the fluid nodes approach the structure, and wherever the distance is smaller than the critical one, the fluid elements are created.

Thus, in the case of the Lagrangian framework used for both the fluid and the structure, this "interaction" consists exclusively in interchanging the boundary conditions applied directly at the matching interface mesh nodes. Next we should address the partitioned coupling strategy associated

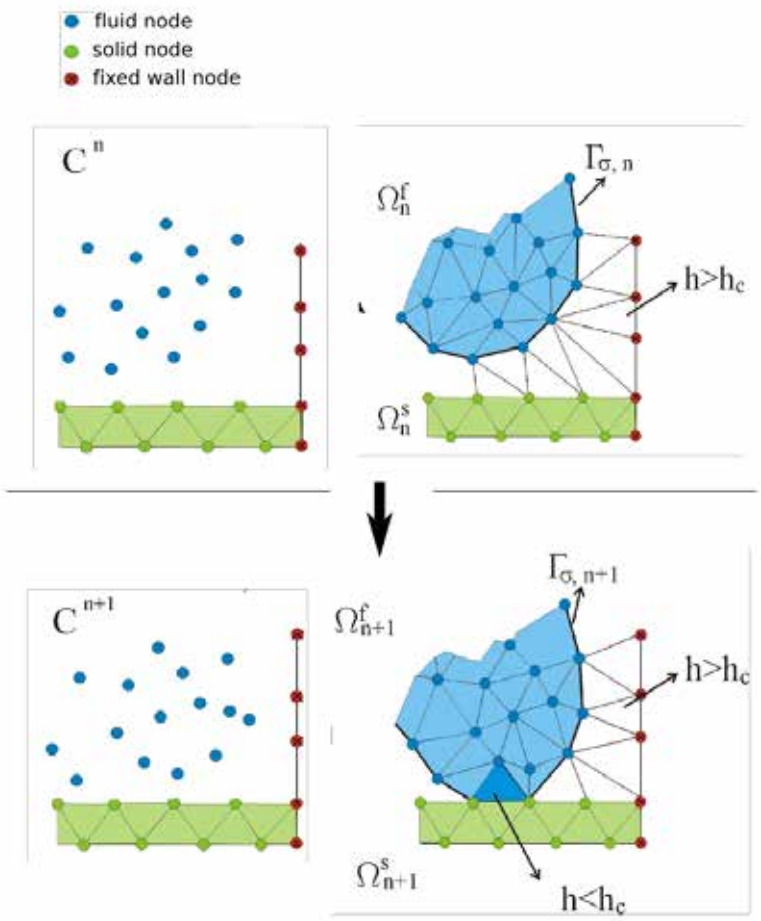


Figure 2.6: Detecting fluid-structure contact in PFEM [3]

with the Lagrangian-Lagrangian setting in the context of the problem of interest, namely the sea-landing of satellite vehicle.

Problem of interest The motivation of this part of this work is the modeling the sea landing of a satellite vehicle. With this application in mind we derive next the model for the vehicle, keeping in mind that the fluid will be modeled by PFEM described above. After that we shall proceed with description of the coupling strategy and the resulting FSI solver for the sea-landing simulation. .

2.4 Rigid body model

In space-craft dynamics one commonly stands between two choices: the first consists in developing the equations of motion modeling the space-craft as a system of point masses [42] m_i, $i = 1, ..., n$ resulting $3n$ second order differential equations of motion complemented by $n-1$ distance constraints. Another option, that is widely used in practice consists in modeling the spacecraft as a rigid body. This implies that rotational degrees of freedom need to be taken into account. This leads to some implications related to the frame of reference chosen for formulating the governing equations of motion: Newtons' second law in the body-fixed rotating reference frame obtains a form different from the habitual (since rotating reference frames are not inertial). However only 6 equations describe the motion, as the rigid body is usually represented by its center of mass.

The latter, namely the rigid body model is chosen in this work as for numerical simulation different discretizations would require re-computation of mass-points distributions and weights if the mass-point system was chosen. In contrast for the rigid body model this is not the case as the key features are the moments of inertia of the body that do not alter with the discretization.

Rigid body governing equation A rigid body is a model used for solid bodies of finite size in which deformation is neglected. In other words, the distance between any two given points of a rigid body remains constant in time regardless of external forces exerted on it. In practice this simplification is chosen when the expected deformations do not affect the motion of the system and the associated stresses are of no interest. Rigid bodies are completely defined by

- position of the center of mass
- mass
- moments of inertia tensor

The motion of a rigid body can be characterized by six degrees of freedom (translation in three directions and rotation about the three axes). The equations describing the motion of a rigid body can be classified into two groups: equations of translational motion and equations of rotation.

We shall start with the equations describing translation. First, let us define the necessary quantities.

The total mass of a rigid body is computed as:

$$m = \int_{\mathcal{B}} dm = \int_{\mathcal{B}} \rho dV \tag{2.49}$$

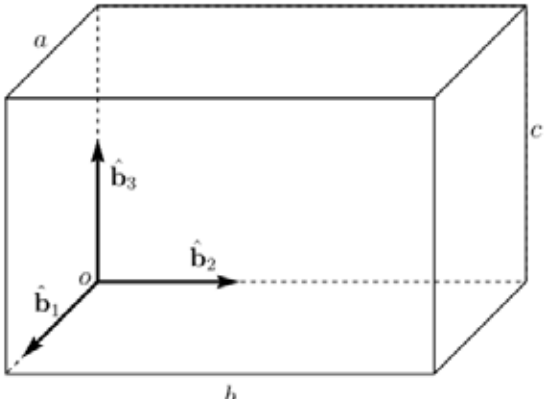


Figure 2.7: A parallelepiped for illustration of mass and moment of inertia calculation

where ρ is the distributed mass (density), that can be both a function of space and time, however it is usually assumed to be constant for most of the practical problems.

The **first moment of inertia S** is defined as an integral of the radius-vector:

$$\mathbf{S}_o = \int_{\mathcal{B}} \mathbf{r}_o dm \tag{2.50}$$

This quantity depends on the origin from which $\mathbf{r}$ is measured, which motivates the use of the subscript "o" to denote that $\mathbf{r}_o$ is measured from "o", that $\mathbf{S}_o$ is the first moment about "o" (see Fig. 2.7). The first moment of inertia is used in practice for the determination of the centroid or the ***center of mass***, a point c around which the first moment of inertia is zero: $\mathbf{S}_c = 0$.

At this point we have introduced the quantities necessary for development of the equations of translational motion of a rigid body.

Rotating reference frames In the following we shall formulate the governing equations. Two options with respect to reference frames are usually followed. First one consists in writing the equations of motion in the inertial (space fixed) reference frames $\mathcal{F}_i$, where Newtons' laws have their classical form. Another option is to write the equations of motion in a body-fixed reference frame[10] $\mathcal{F}_b$. We follow the second option, as it is more convenient to write the rotational equations in a body-fixed frame as we shall see later.

The relationships between the main kinematic quantities in the inertial and rotating body-fixed reference frames are as follows (indices "i" and "b" stand for "inertial" and "body" reference frames respectively):

[10]Note that the body-fixed reference frame in the presence of rotation is not inertial.

$$\mathbf{v}_i = \mathbf{v}_b + \boldsymbol{\omega} \times \mathbf{r} \tag{2.51}$$

(where $\mathbf{v}_i$ and $\mathbf{v}_b$ are the velocities in the inertial and rotating reference frames respectively, $\mathbf{r}$ is the radius vector from the inertial origin to the origin of the rotating reference frame and $\boldsymbol{\omega}$ is the angular velocity of the body-fixed frame with respect to the inertial reference frame (see Fig. 2.8).

$$\mathbf{a}_i = \mathbf{a}_b + 2\boldsymbol{\omega} \times \mathbf{v}_b + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) - \frac{d\boldsymbol{\omega}}{dt} \times \mathbf{r} \tag{2.52}$$

and the quantities $\mathbf{v}_b$ and $\mathbf{a}_b$ are defined as

$$\mathbf{v}_b = \left(\frac{d\mathbf{r}}{dt}\right)_b \tag{2.53}$$

$$\mathbf{a}_b = \left(\frac{d^2\mathbf{r}}{dt^2}\right)_b \tag{2.54}$$

namely the time derivatives taken in the rotating (body-fixed) reference frames.

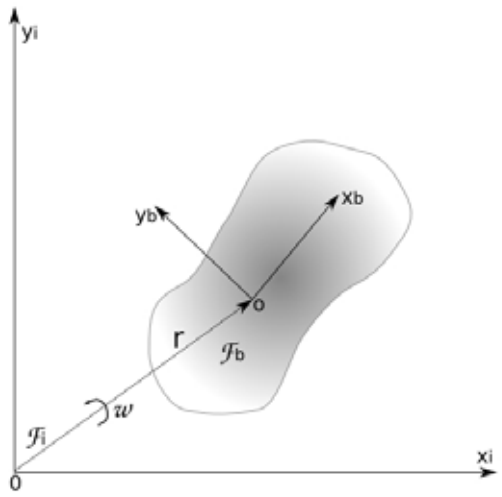


Figure 2.8: Inertial and body-fixed reference frames

Equations of a rigid body for translational motion The linear momentum of a rigid body is defined as

$$\mathbf{p} = m\mathbf{v}_o \tag{2.55}$$

where $\mathbf{v}_o$ is the velocity of the gravity center of the rigid body (see Fig. 2.8).

Now we shall extend the second law of Newton $\sum \mathbf{f}_{ext} = \partial_t \mathbf{p}$ to the motion of rigid bodies.

Using Eq.(2.51) we obtain the governing equation of translational motion, written in the body-fixed reference frame with the origin placed in the gravity center of the rigid body:

$$\mathbf{f} = \partial_t \mathbf{p} + \boldsymbol{\omega} \times \mathbf{p} = m\partial_t \mathbf{v}_o + \boldsymbol{\omega} \times (m\mathbf{v}_o) \tag{2.56}$$

Rotational equations of a rigid body motion Now, the rotational equations of motion for the rigid body shall be presented. The clearest way is based upon Euler's law, stating that the change of angular momentum around the mass center is equal to the net applied moment about the mass center. We write it first in the classical inertial frame form as:

$$\partial_t \mathbf{h} = \mathbf{Q} \tag{2.57}$$

where $\mathbf{h}$ is the angular momentum about the mass center and $\mathbf{Q}$ is the total applied moment about the mass center.

Angular momentum of the rigid body in a rotating reference frame is defined as

$$\mathbf{h} = \int_{\mathcal{B}} \mathbf{r} \times (\mathbf{v}_o + \boldsymbol{\omega} \times \mathbf{r})\, dm \tag{2.58}$$

In the body-fixed reference frame located at the gravity center it can be expressed [11] as $\mathbf{h} = \mathbf{I}\boldsymbol{\omega}$, where $\mathbf{I}$ is the moment of inertia tensor (constant in the body-fixed reference frame, which is one of the reasons it is convenient to write the equations of motion in the body fixed frame) and $\boldsymbol{\omega}$ - the angular velocity of the body-fixed reference frame $\mathcal{F}_b$ with respect to the room-fixed frame $\mathcal{F}_i$. [12]

[11] By definition angular momentum of a rigid body in a rotating reference frame is

$$\mathbf{h} = \int_{\mathcal{B}} \mathbf{r} \times (\mathbf{v}_o + \boldsymbol{\omega} \times \mathbf{r})\, dm = \int_{\mathcal{B}} \mathbf{r} \times \mathbf{v}_o dm + \int_{\mathcal{B}} \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r})\, dm \tag{2.59}$$

Using the triple cross product rule, namely $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}\,(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}\,(\mathbf{a} \cdot \mathbf{b})$ we can express Eq. (2.59) as

$$\mathbf{h} = \int_{\mathcal{B}} \mathbf{r} \times \mathbf{v}_o dm + \int_{\mathcal{B}} [\mathbf{w}\mathbf{r} \cdot \mathbf{r} - \mathbf{r}\,(\mathbf{w} \cdot \mathbf{r})]\, dm \tag{2.60}$$

Introducing the 3x3 identity tensor $\mathcal{I}$ and noticing that $\mathbf{v}$ and $\boldsymbol{\omega}$ do not depend on the differential mass element Eq. (2.60) can be rewritten as

$$\mathbf{h} = \int_{\mathcal{B}} \mathbf{r} dm \times \mathbf{v} + \boldsymbol{\omega} \cdot \int_{\mathcal{B}} \left(\mathbf{r}^2 \mathcal{I} - \mathbf{r} \otimes \mathbf{r}\right) dm \tag{2.61}$$

The first integral is the first moment of inertia. If we choose the gravity center of the rigid body as the origin, it vanishes and thus the angular momentum simplifies to

$$\mathbf{h} = \boldsymbol{\omega} \cdot \int_{\mathcal{B}} \left(\mathbf{r}^2 \mathcal{I} - \mathbf{r} \otimes \mathbf{r}\right) dm \tag{2.62}$$

[12] If the room-fixed frame is chosen to describe the motion, the inertia tensor must be updated at every step by applying the incremental rotation matrix.

The second moment of inertia $\mathbf{I}$ is a tensor quantity defined as a volume integral:

$$\mathbf{I} = \int\int\int_{\mathcal{B}} \rho \left(\mathbf{r}^2 \mathcal{I} - \mathbf{r} \otimes \mathbf{r} \right) dxdydz \tag{2.63}$$

with $\mathcal{I}$ being a 3x3 identity tensor. Moment of inertia is a measure of an object's resistance to changes in its rotation rate. It can be understood as an equivalent of mass for the rotational motion.

In a rotating reference frame $\mathcal{F}_b$ Eq. (2.57) obtains the form:

$$\partial_t \mathbf{h} + \boldsymbol{\omega} \times \mathbf{h} = \mathbf{Q} \tag{2.64}$$

Since $\mathbf{h} = \mathbf{I}\boldsymbol{\omega}$ and $\partial_t \mathbf{I} = 0$ in the body-fixed frame the Eq. (2.64) can give us directly an expression for the angular acceleration:

$$\dot{\boldsymbol{\omega}} = -\mathbf{I}^{-1}\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \mathbf{I}^{-1}\mathbf{Q} \tag{2.65}$$

If the principle reference frame is used, then the moment of inertia is diagonal and the the matrix equation (2.65) can be expanded to obtain the "standard" version of Euler's equations for the rotational motion of a rigid body:

$$\dot{\boldsymbol{\omega}}_1 = \frac{I_2 - I_3}{I_1}\boldsymbol{\omega}_2\boldsymbol{\omega}_3 + \frac{Q_1}{I_1} \tag{2.66}$$

$$\dot{\boldsymbol{\omega}}_2 = \frac{I_3 - I_1}{I_2}\boldsymbol{\omega}_1\boldsymbol{\omega}_3 + \frac{Q_2}{I_2} \tag{2.67}$$

$$\dot{\boldsymbol{\omega}}_3 = \frac{I_1 - I_2}{I_3}\boldsymbol{\omega}_1\boldsymbol{\omega}_2 + \frac{Q_3}{I_3} \tag{2.68}$$

$$\tag{2.69}$$

where indices $1, 2, 3$ refer to the directions of the principle axes of rotation.

Finally, the set of equations describing the motion of rigid body consists of translational equations (2.56) (three scalar second order differential equations in position, one for each spatial acceleration component) and three equations for the rotational motion Eq.(2.65). Determination of the primary variables (positions and rotation angles) is obtained by integrating these equations in time. In the current work the time integration of the rigid body equations was carried out by a standard Newmark-Bossak scheme[13] in order facilitate the direct use of the existing code. The integrating formulae for this scheme are presented in **Chapter 3**. Note that for time integration of the rigid-

[13]The use of standard time integration formulas such as Newmark-Bossak is justified for small time steps, which is valid for the present case as the time step chosen for the integration of the structural problem in an FSI context in this work is taken to be equal to the one for the fluid.

body equations in presence of finite rotations there exist specialized quaternion-based schemes [43], [44].

Incremental rotation matrix As for now we have defined a set of governing rigid body equations whose solution and integration (in time) shall provide us with the kinematic quantities of the rigid body.

We have decided to write the dynamic equations in the the body-fixed reference frame in order not to recompute the moments of inertia. The "catch" however consists in the necessity in such a case to transform the external forces and moments (that are usually given in the room-fixed inertial reference frame) into the rotating body-fixed frame. This requires definition of a transformation matrix, mapping a vector or tensor quantity between $\mathcal{F}_b$ and $\mathcal{F}_i$. From a first glance it seems to be an easy procedure that simply requires the knowledge of the angles between the current position of the body fixed reference frame and the room fixed reference frame and application of the resulting transformation matrix to the given vector. This however is not the case.

The difficulty arises from the fact that 3D rotations are generally not commutative, namely that if the body undergoes a sequence of rotations $\theta_1 ... \theta_n$ around different axes the total rotation $\theta_{tot} \neq \theta_1 + ... \theta_n$. The finite rotations are commutative if and only if they were performed around one axis (that is the rotations in one plane). In his essential work Argyris [45] recognized the complex nature of 3D rotations. The investigation went on further in the work of Simo and Vu-Quoc [46], and finally in the extensive and rigorous analysis of Ibrahimbegovic [47]. 3D rotations define a complicated topic by itself and its analysis lies outside of the current work' scope. Here we restrict ourselves merely to the results as they can be found in the literature. The "correct" form of computing the rotation matrix is the incremental way. Namely the rotation matrix at time t_n must be computed as a product of incremental rotation matrices[14]:

$$\mathbf{R}_n^{total} = \mathbf{R}_n^{incr} * \mathbf{R}_{n-1}^{total} = \mathbf{R}_n^{incr} * \mathbf{R}_{n-1}^{incr} * \mathbf{R}_{n-2}^{incr} * ... \mathbf{R}_1^{incr} \tag{2.70}$$

with the incremental rotation matrix $\mathbf{R}_n^{incr}$ (corresponding to the rotations increments θ_x, θ_y and θ_z at time n about the room-fixed axes x, y and z) defined as

[14] Note that in the context of a linearization, the rotation matrix is updated inside of the non-linear loop, i.e. then the "incremental" angles correspond to the position update associated to every non-linear iteration.

$$\mathbf{R}_n^{incr} = \begin{pmatrix} cos\theta_z cos\theta_z + sin\theta_z sin\theta_x sin\theta_y & sin\theta_z cos\theta_x & -cos\theta_z sin\theta_y + sin\theta_z cos\theta_x cos\theta_y \\ -sin\theta_z cos\theta_y + cos\theta_z sin\theta_x sin\theta_y & cos\theta_z cos\theta_x & sin\theta_z sin\theta_y + cos\theta_z sin\theta_x cos\theta_y \\ cos\theta_x sin\theta_y & -sin\theta_x & cos\theta_x cos\theta_y \end{pmatrix} \tag{2.71}$$

Using the rotation matrix we can obtain the necessary transformation between $\mathcal{F}_b$ and $\mathcal{F}_i$. Thus a vector $\mathbf{a}$ in the room-fixed reference frame turns into $\mathbf{R}_n^{total}\mathbf{a}$ in the body-fixed reference frame at time t_n.

The skin The "skin" of a rigid body in the context of a numerical implementation is a set of the surface nodes resulting from the spatial discretization. Location of a node A of a rigid body is determined by a vector $\mathbf{r}_A^b$ in the body fixed reference frame (which does not change). However the position of the node A in the room fixed reference frame is $\mathbf{r}_A^i = \mathbf{r}_c^i + \mathbf{R}^{total}\mathbf{r}_A^b$ (note that super-indices "i" and "b" are used to distinguish between the reference frames). The first term is the position of body-fixed frame origin usually placed at the mass center of our rigid body. The second term reflects the above-explained rotation transformation.

The computation of the variables at the nodes of the rigid body "skin" is essential for the interaction, as it permits definition of the "external forces" at nodes acting upon the fluid at contact.

2.5 Strongly coupled FSI involving an incompressible fluid and a rigid body

The most classical way of coupling the fluid and the solid in a partitioned FSI approach is based upon the Dirichlet-Neumann coupling, which supplements the fluid sub-problem with Dirichlet boundary conditions (ensuring continuity of velocities) and the structural problem with Neumann boundary conditions (ensuring continuity of stresses). Generally, Dirichlet-Neumann couplings work well as long as one deals with solids undergoing small deformations and for large density ratios between the solid and the fluid.

In this chapter we aim at modeling the interaction between water and the rigid body (dictated by the application of interest) and thus as a starting point Dirichlet-Neumann coupling is chosen.

The algorithm for the standard Dirichlet-Neumann FSI coupling in conjunction with the introduced fractional step method can be written as follows:

1. Solve the equations of motion for the structure for displacement $\bar{\mathbf{u}}^0_{n+1}$, using the the fluid pressure $\bar{p}_n$
2. Map the displacements to the fluid domain and deform the mesh accordingly
3. Solve fractional momentum equation for $\tilde{\mathbf{v}}$ using $\bar{p}_n$
4. Start loop until convergence in structural displacement
   Solve the fluid pressure equation (second fractional step) for $\bar{p}^i_{n+1}$
   Solve the structural equations for $\bar{\mathbf{u}}^i_{n+1}$ using $\bar{p}^i_{n+1}$
   Solve the end-of-step momentum equation for the fluid (last fractional step)

Table 2.2: Standard Dirichlet-Neumann coupling for the FSI with a fractional step

The coupling problems occur in this setting as in the second stage of the fractional step method (calculation of pressure), the effect of structural motion upon the fluid is ignored. The idea of improving the Dirichlet-Neumann coupling consists in modifying the second stage of the fractional step scheme (solution of the pressure equation) so to account for the effect of the velocity of the structure upon the fluid pressure [20]. The modified second fractional step can be written as (we omit stabilization terms for the sake of simplicity)

$$\mathbf{D}\tilde{\mathbf{v}} = \hat{\mathbf{L}}\left(\bar{p}_{n+1} - \bar{p}_n\right) \tag{2.72}$$

where the modified Laplacian is defined as

$$\hat{\mathbf{L}} = \mathbf{L} + \mathbf{L}_{fs} = \mathbf{D}\mathbf{M}^{-1}\mathbf{G} + \mathbf{D}_{fs}\mathbf{M}^{-1}_{fs}\mathbf{G}_{fs} \tag{2.73}$$

$\mathbf{D}_{fs}$, $\mathbf{G}_{fs}$ and $\mathbf{M}_{fs}$ stand for the divergence, gradient and mass matrices corresponding to the fluid-structure interface. In case of the rigid body the mass matrix is defined by associating to every point of the rigid body skin a fraction of the total mass corresponding to the total number of nodes. Thus, the introduced interface Laplacian $\mathbf{L}_{fs} = \mathbf{D}_{fs}\mathbf{M}^{-1}_{fs}\mathbf{G}_{fs}$ accounts for the effect of the structural motion (inertia) upon the fluid.

The modification introduced assures convergence, while without it the solution is found to be instable. The detailed derivation of the introduced technique on the basis of the original monolithic FSI system as well as additional modifications in the case of flexible structures are given in [20].

2.6 Examples

This section deals with an industrial problem that motivated the developments of the chapter. Even though the project required a large number of simulations we shall present just some selected results.

The objective was to model the impact of the satellite capsule (see Fig. 2.9) against the sea surface as well as the subsequent floating of the capsule. The aim of the first phase was to estimate the forces acting upon the capsule during the impact and to get qualitative insight of the motion of the capsule for different angles of attack. For the analysis of the impact, floats were not considered. The vehicle is designed in a way that the floats get inflated after the impact, i.e. at the moment when danger of sinking appears.

The second phase was devoted to the analysis of vehicle's floating, when the security balloons inflate to keep the vehicle on the sea surface. The deformation of the floats was neglected and they, as well as the vehicle were modeled as a rigid body.

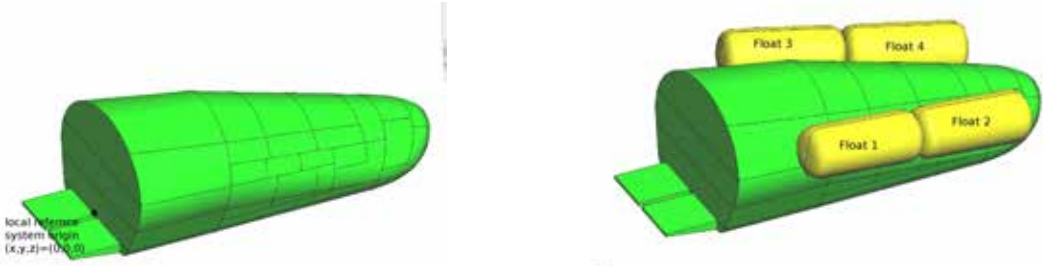
Properties of the capsule For all the analysis cases below the following properties of the vehicle were taken:

- Volume $V = 7.70[m^3]$
- Mass $m = 1710[kg]$
- Center of gravity: (1.822,-0.0039,-0.1179)
- Inertia tensor: I=

492	0.0	0.0
0.0	2247	0.0
0.0	0.0	2358

The off-diagonal moments were negligible. The body-fixed reference frame system was located at the mass center of the capsule. The position of the mass center of the capsule was counted from the point indicated in Fig. 2.9.

Water was modeled as an incompressible fluid using the PFEM. The fluid domain had a half-cylindrical shape, with length of 10 [m] in the x direction, and a 180 degrees circular segment cross-section with a radius of 5 [m]. Water was discretized by 800.000 linear tetrahedra (if not otherwise specified).



(a) Capsule (b) Capsule with floats

Figure 2.9: A capsule, floats numbering and origin of the local coordinate system

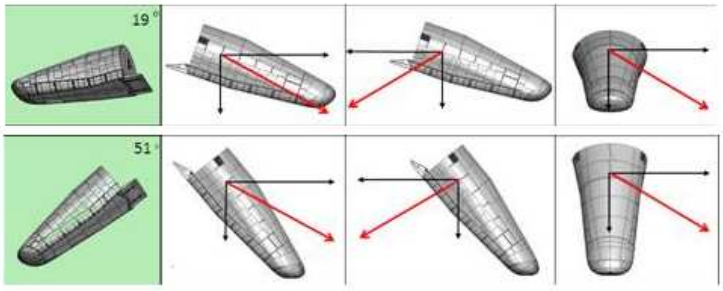


Figure 2.10: Impact scenarios

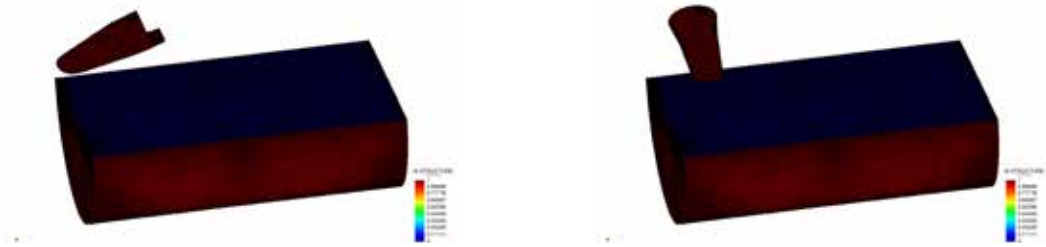
2.6.1 Impact simulation

The first analysis concerned with the modeling of the impact of the vehicle against the initially still water surface. At this stage the possibility of water entering the capsule was neglected, and thus the possible alterations of the inertial properties of the vehicle due to accumulation of water inside were neglected. The main point of interest was the estimation of the maximum accelerations/forces exerted upon the vehicle during the impact as well as the stability of the vehicle (possibility of the flip-over). The following impact scenarios were simulated: the angles between the longitudinal axis of the satellite and the water surface for the simulations were chosen to be 19 and 51 degrees. The initial descent velocities studied were $v_{x0} = 10$ [m/s] and $v_{z0} = -6$ [m/s]. At time t=0.0 [s] the capsule was situated at z_{init}=3 [m] above the water surface. The frontal, rear and lateral impacts were considered. These are indicated by the red arrows in the Fig. 2.10.

Out of 6 cases analyzed, here we present the results of two cases only, namely

- Lateral impact, angle of attack 19 degrees: **Lat19**
- Rear impact, angle of attack 19 degrees: **Rear19**

The initial position of the capsule for these cases is presented in Fig. 2.11.



(a) Rear19 (b) Lat19

Figure 2.11: Initial position of the capsule: 2 analysis cases

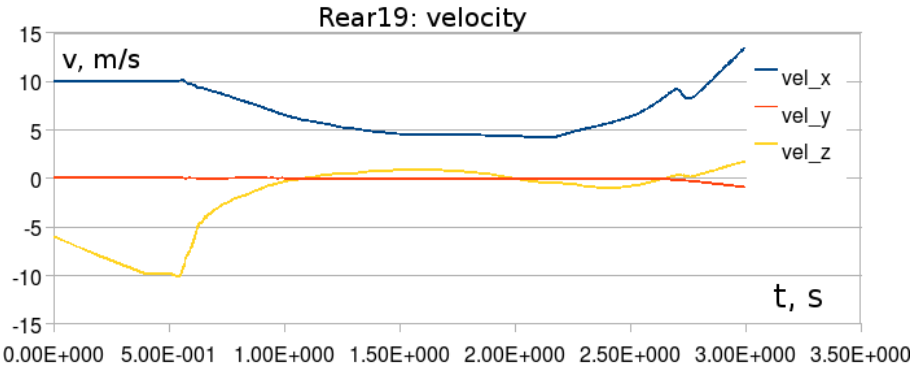


Figure 2.12: **Rear19**: velocity

Rear19 A series of snap-shots, taken at t=0.0, 0.6, 1.0, 1.4, 1.8, 2.2 [s] is presented in Fig. 2.14. Impact occurs roughly at t=0.5 [s], that is at the point of time when both the horizontal and vertical velocities are v_x=10 [m/s] and vertical velocity v_y=-10 [m/s] (see Fig. 2.12). A large angular velocity ω_y gets developed reaching its highest negative value $\omega_y \approx$-2 [rad/s] at t=0.6 [s]. Angular velocities in ω_x and ω_z are rather negligible as the impact is symmetric. At t=0.8 [s] a maximum positive angular velocity ω_y of $\approx$ 2 [rad/s] is developed. This can be seen in Fig. 2.13, where the temporal evolution of angular velocities measured in the room-fixed reference frame is presented.

Figures 2.16 and 2.16 depict the rotations and translations measured in the global reference frame for the case **Rear19**. At t$\approx$1.5 [s] the sense of rotation around y axis changes. The case **Rear19** is symmetric in the XZ plane.

Fig. 2.17 depicts the accelerations and angular accelerations measured in the global reference frame for the case **Rear19**. The maximum acceleration of the capsule during the impact is reached between 0.5 and 0.6 [s].

In Fig. 2.17 one can see high frequency fluctuations of acceleration. Apparently, the time step used in the simulation was sufficiently small to resolve those. However we believe that these can be

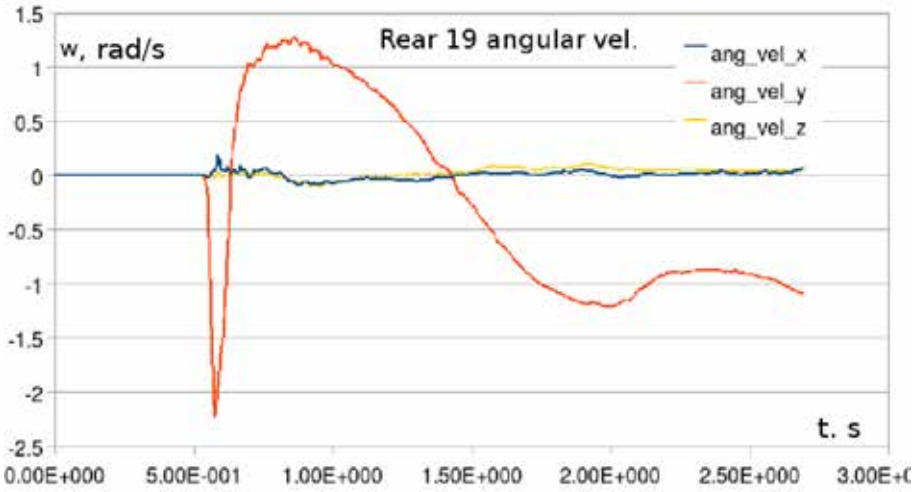
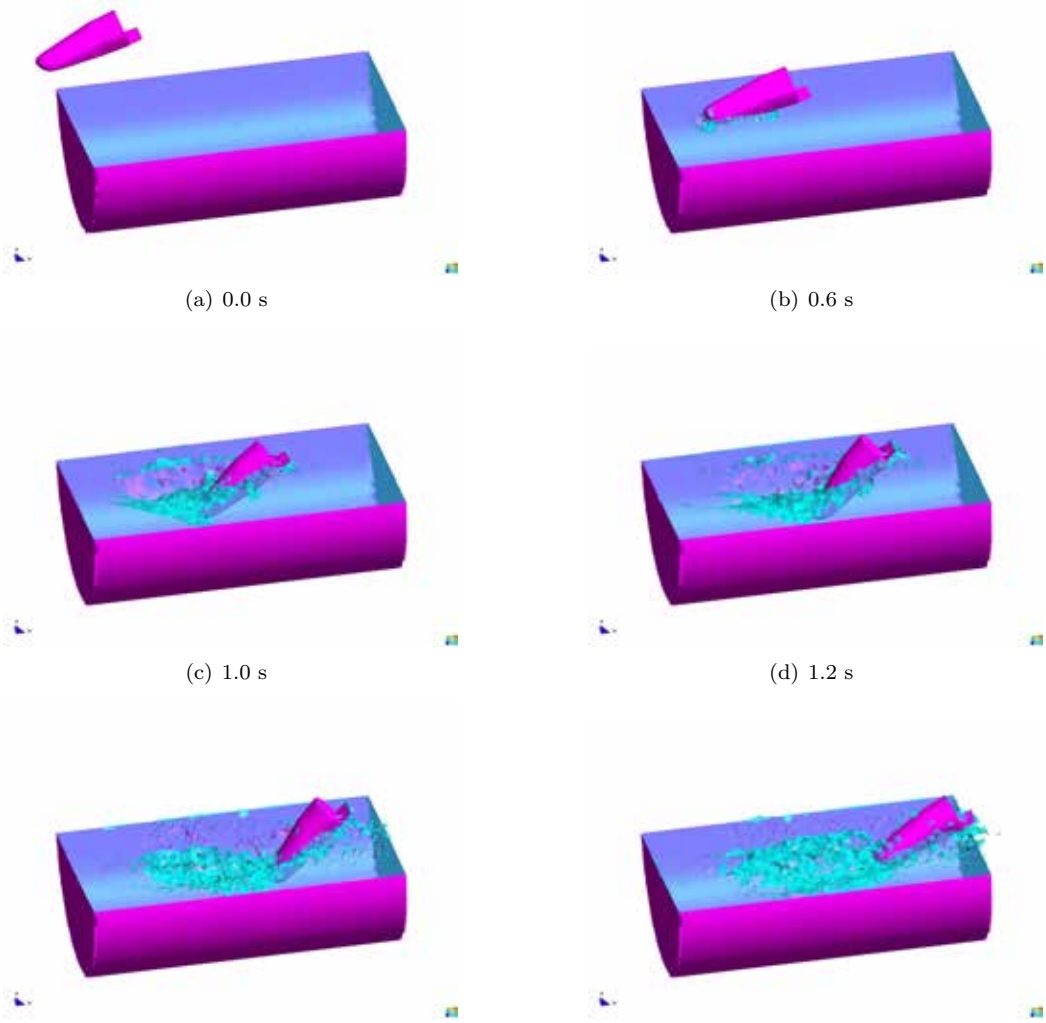


Figure 2.13: **Rear19**: angular velocity

considered "numerical noise". The only way to prove this, is the comparison with the measurements, which are unfortunately not available.

Lat19 In the case of lateral impact flip-over takes place, rendering this case as potentially dangerous. This is depicted in a series of snap-shots, taken at t=0.0, 0.5, 1.0, 1.4, 1.8, 2.2 [s] and shown in Fig. 2.18.



(a) 0.0 s (b) 0.6 s

(c) 1.0 s (d) 1.2 s

(e) 1.6 s (f) 2.0 s

Figure 2.14: **Rear19** case at six time instances

Figure 2.15: **Rear19**: rotations

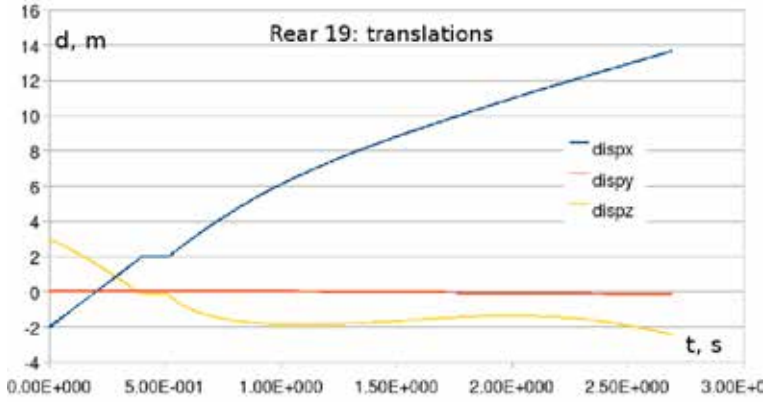


Figure 2.16: **Rear19**: displacements

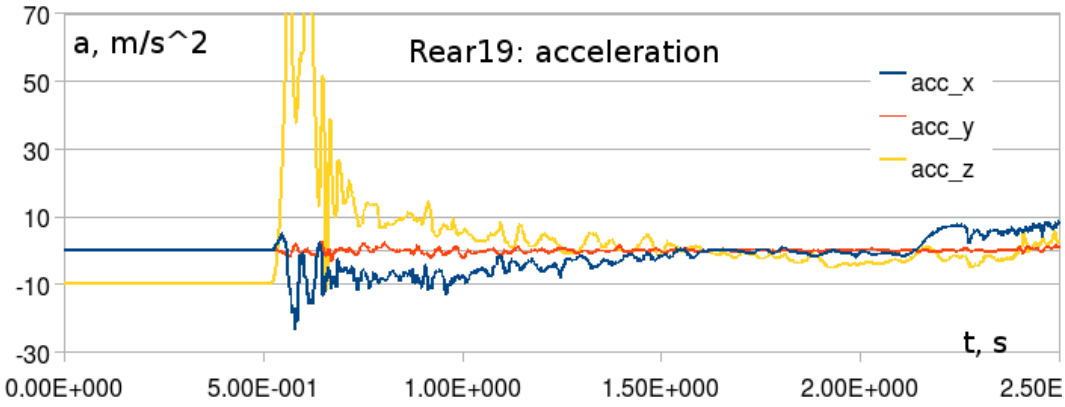
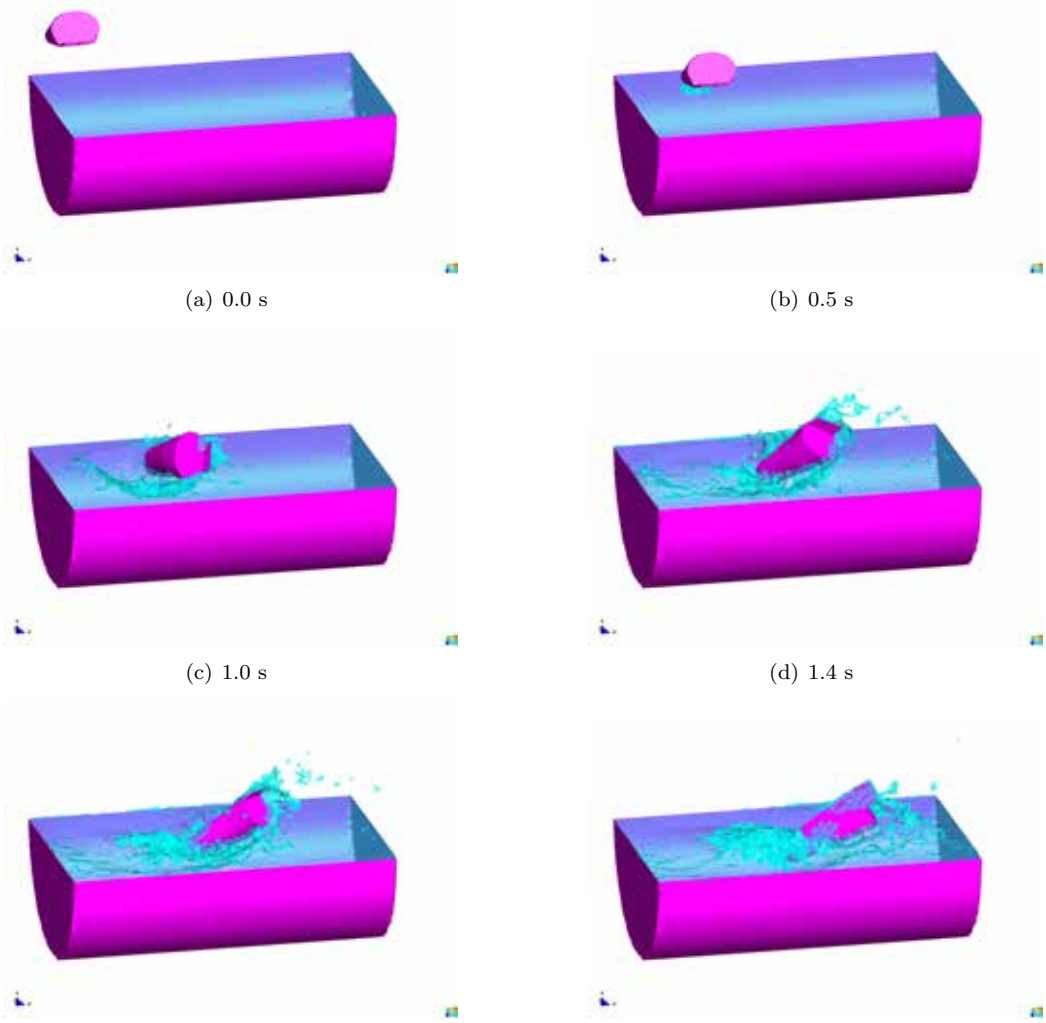


Figure 2.17: **Rear19**: acceleration



(a) 0.0 s (b) 0.5 s

(c) 1.0 s (d) 1.4 s

(e) 1.8 s (f) 2.2 s

Figure 2.18: Flip-over of due to impact for the case **lat19**

Next, the velocity evolution is presented in Fig. 2.19. Impact starts at around 0.6 seconds, with velocities v_x ≈10 [m/s], v_y ≈ 10 [m/s]. One can see, that the at the moment of impact the velocity component v_y that was zero initially begins to develop and becomes zero again once the flip-over is complete. Fig. 2.20 shows the temporal evolution of the rotations. One can see that at the end of the simulation, the capsule turned around 280 degrees around the global vertical axis **z**. This corresponds to 5.5 [rad]. Maximal angular velocities ω_y^{max} ≈5 [rad/s] and ω_z^{max} ≈-2 [rad/s] were exhibited at around t=0.75 [s]. This is presented in Fig. 2.21 Figures 2.22 and 2.23 depict temporal evolution of the displacement and accelerations of the capsule. Maximal vertical acceleration due to impact of ≈ $70\,m/s^2$ is exhibited at t=0.65 [s]. These data lie within the expected range, according to the experiments performed by the industrial project partner CIMSA Ingeneria de Sistemas, S.A.

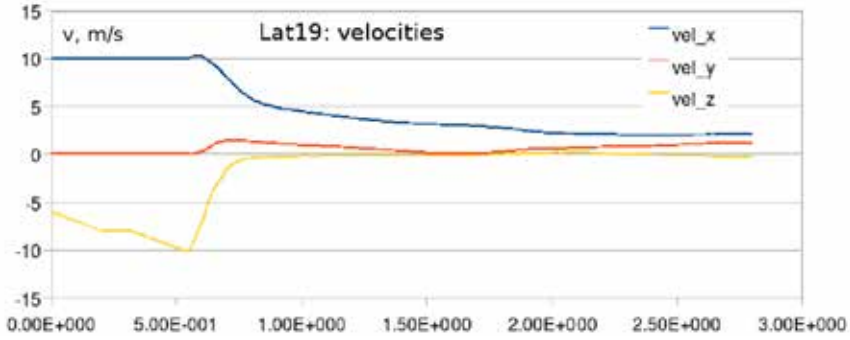


Figure 2.19: Lateral impact: velocities

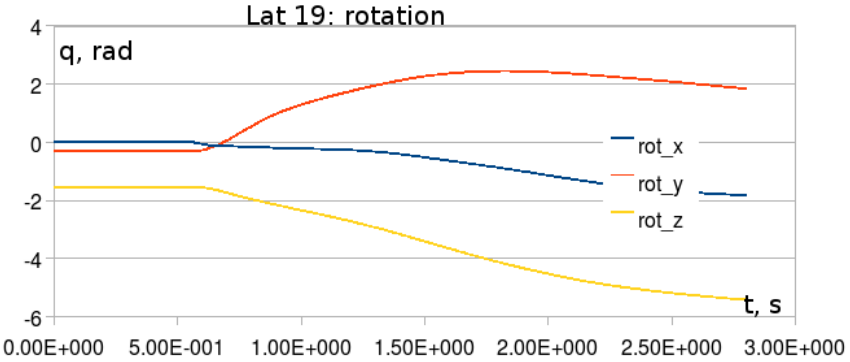


Figure 2.20: Lateral impact: rotations

2.6.2 Floatability analysis

A floatability analysis of the capsule equipped with inflatable balloons was also performed. Here, the possibility of water penetration was included. The aim of this analysis was to

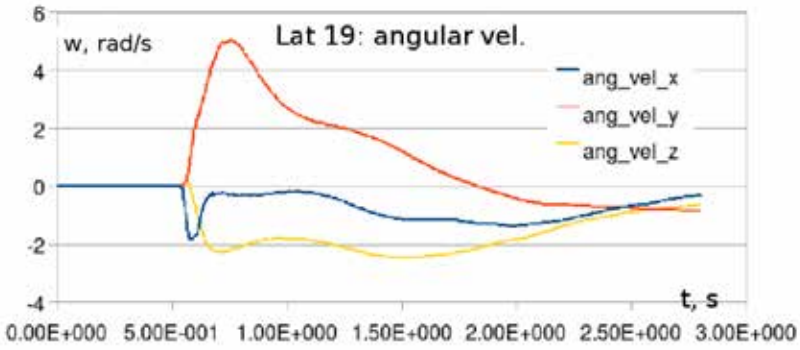


Figure 2.21: Lateral impact: angular velocities

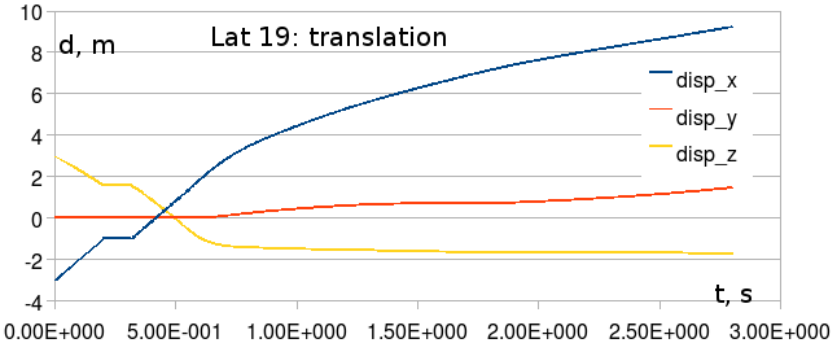


Figure 2.22: Lateral impact: translation

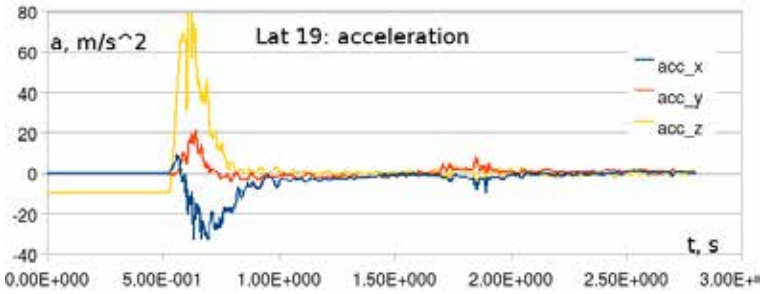


Figure 2.23: Lateral impact: acceleration

1. check if the volume of the designed inflatable floats was enough to sustain the capsule at the water surface for different locations of water accumulation in the capsule

2. find the equilibrium position of the capsule

3. find the forces acting upon the floats

Each float was connected to the capsule at four attachment points modeled as rigid connections. The leakage was not simulated directly, instead, the critical volume of water that can enter into the capsule was estimated and taken into account by appropriate changes of the mass, center of gravity and the moments of inertia of the capsule. Calculation of the inertia tensor of the capsule with water requires re-computation of the empty capsule inertia tensor with respect to new gravity center and addition of the moment of inertia of the accumulated water. The center of gravity of the entire system is computed as:

$$\mathbf{r}_{glob} = \frac{\sum_i \mathbf{r}_i m_i}{\sum_i m_i} \tag{2.74}$$

where $\mathbf{r}$ is the radius-vector pointing to the gravity center.

Next, given the inertia properties of each sub-entity (empty capsule, water) with respect to its center of gravity the inertias with respect to the global center of gravity $\mathbf{r}_{glob}$ were found and added up: let us assume that the entire system (capsule filled with water) consists of an empty capsule with inertia tensor $\mathbf{I}^{loc}_{caps}$ with the gravity center $\mathbf{r}_{caps}$ and water inside with inertia tensor $\mathbf{I}^{loc}_{wat}$ with the gravity center $\mathbf{r}_{wat}$. The gravity center of the entire system is found according to Eq. (2.74) and will be denoted $\mathbf{r}_{glob}$. The inertia tensor of the entire system with respect to its center of gravity $\mathbf{r}_{glob}$ is found as a sum of the inertia tensors of an empty capsule and water, each one written with respect to the new center of gravity. According to Steiner's theorem, this gives finally:

$$\mathbf{I}_{total} = \mathbf{I}^{loc}_{caps} + m_{caps}\left(\mathbf{R}_1 \cdot \mathbf{R}_1 \mathcal{I} - \mathbf{R}_1 \otimes \mathbf{R}_1\right) + \mathbf{I}^{loc}_{wat} + m_{wat}\left(\mathbf{R}_2 \cdot \mathbf{R}_2 \mathcal{I} - \mathbf{R}_2 \otimes \mathbf{R}_2\right) \tag{2.75}$$

where $\mathbf{I}_{total}$ is the inertia matrix of the entire system, $\mathbf{R}_1 = \mathbf{r}_{glob} - \mathbf{r}_{caps}$ and $\mathbf{R}_2 = \mathbf{r}_{glob} - \mathbf{r}_{wat}$, $\mathbf{r}$ being the radius vector, m_{caps} and m_{wat} are the masses of empty capsule and water, and $\mathcal{I}$ is the identity matrix.

Scenarios The capsule with the floats is initially immersed in water as displayed in the Fig. 2.24 below: A detailed view of the inflatable floats is depicted in Fig. 2.25. In order to find the equilibrium position of the capsule in scenarios with water being accumulated at different parts of capsules, 3 configurations were initially chosen for the analysis:

- Water accumulated in the front of the capsule (case **front**)
- Water accumulated at the rear part of the capsule (case **rear**)
- Water accumulated at the lateral of the capsule (case **lat**)

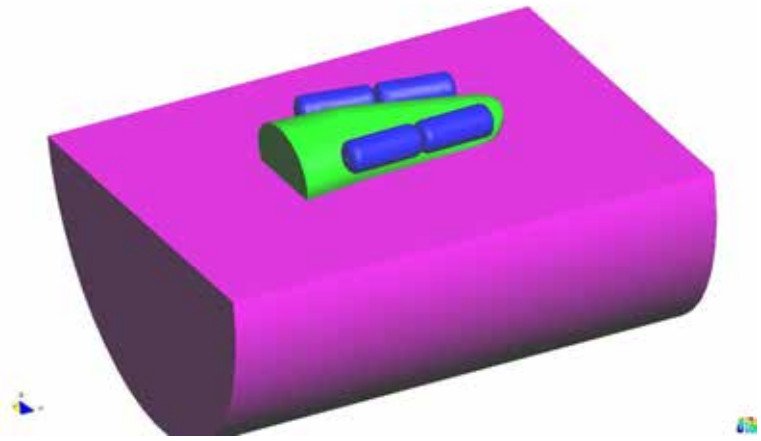


Figure 2.24: Initial configuration of the capsule for the floatability analysis

To estimate the behavior of the capsule in the most unfavorable situation it was assumed that the space between the external panel and internal housing of the capsule is completely filled with water. The properties of the vehicle in this case are:

- Mass $m = 3574[kg]$
- Center of gravity: (1.84, 0, -0.16)
- Inertia tensor: I=

Ixx	Ixy	Ixz
Iyx	Iyy	Iyz
Izx	Izy	Izz

=

1496	0.0	-86.73
0.0	6368	0.0
-86.73	0.0	6873

Additionally for every configuration, water accumulation inside the internal housing was estimated. This leads to following properties to be used in the simulation

Case front:

- Mass $m = 5422[kg]$
- Center of gravity: (2.3, 0, -0.11)
- Inertia tensor: I=

Ixx	Ixy	Ixz
Iyx	Iyy	Iyz
Izx	Izy	Izz

=

1825	0.0	-124.87
0.0	8165.15	0.0
-124.87	0.0	8749

The equilibrium position obtained in this case is depicted in Fig. 2.26.

Fig. 2.27 represents temporal evolution of the cumulative forces acting on first and second floats respectively (due to symmetry of the problem, results for float N3 and N4 are equivalent to those

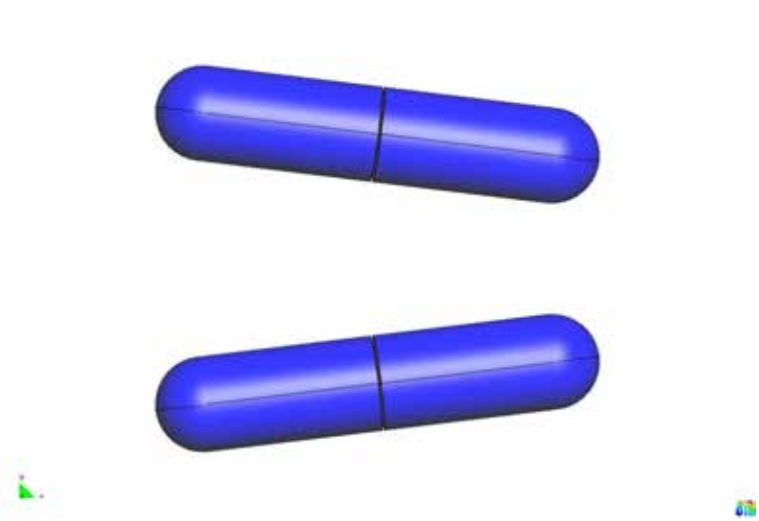
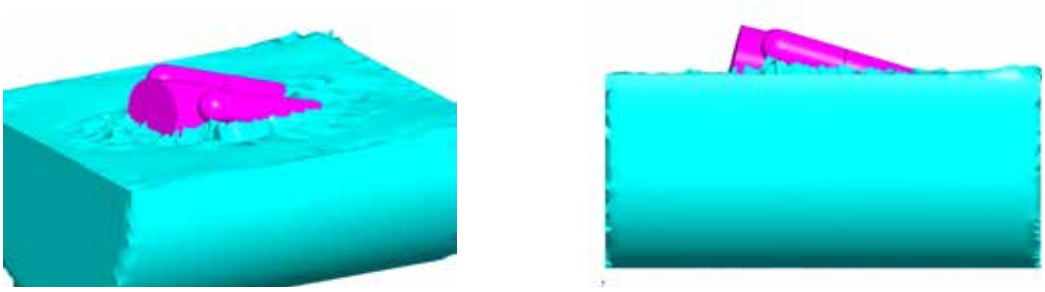


Figure 2.25: Design of inflatable floats



(a) 3D view (b) XZ-plane

Figure 2.26: Equilibrium position of the capsule under the assumption of frontal water accumulation

of N1 and N2 respectively, see Fig. 2.9). Fig. 2.27 shows that under an assumption of water accumulation at the front part of the vehicle, the force is almost entirely applied to the front floats, i.e. Float1, and Float4. The maximum force in vertical direction of value $\mathbf{f}_z$=6 [kN] is exerted upon the front float at ≈1.5 [s]. Maximal forces in x and y directions amount to 1.8 [kN] and 3 [kN] respectively. Forces exerted upon Floats 1 and 3 are negligible.

Case back:

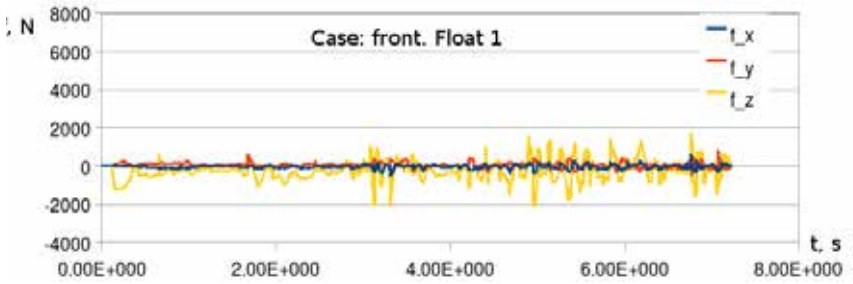
- Mass $m = 6813.5[kg]$
- Center of gravity: (1.39, 0, -0.08)
- Inertia tensor: I=

Ixx	Ixy	Ixz
Iyx	Iyy	Iyz
Izx	Izy	Izz

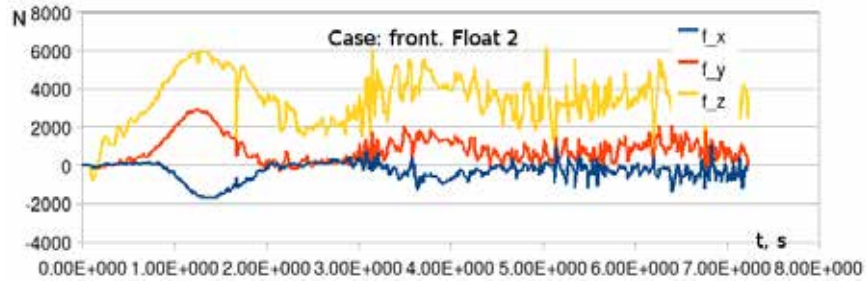
=

2508	0.01	-241
0.0	9061	0.0
-241	0.0	9189

The equilibrium position obtained in this case is depicted in Fig. 2.28. Fig. 2.29 represents

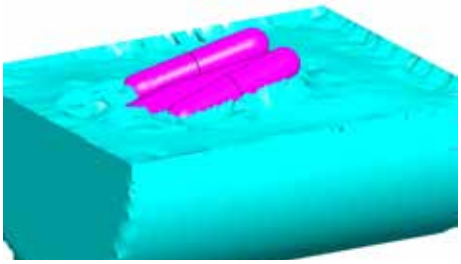


(a) Float 1

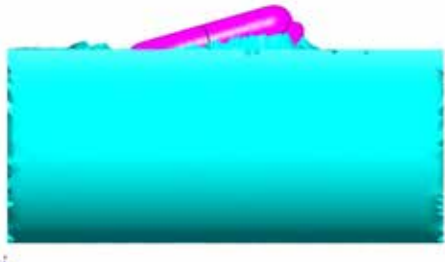


(b) Float 2

Figure 2.27: Temporal evolution of forces acting on the floats under assumption of water accumulation at the front



(a) 3D view



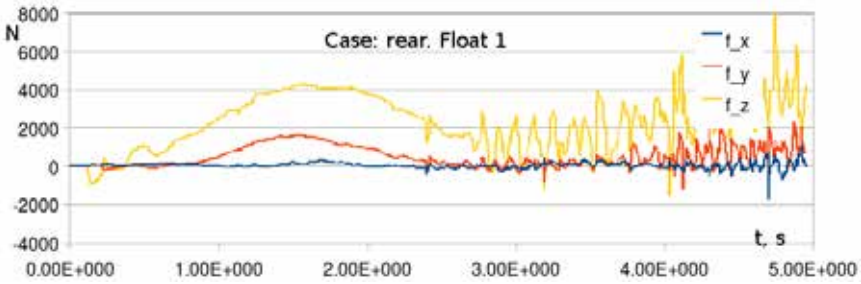
(b) XZ-plane

Figure 2.28: Equilibrium position of the capsule under the assumption of rear water accumulation

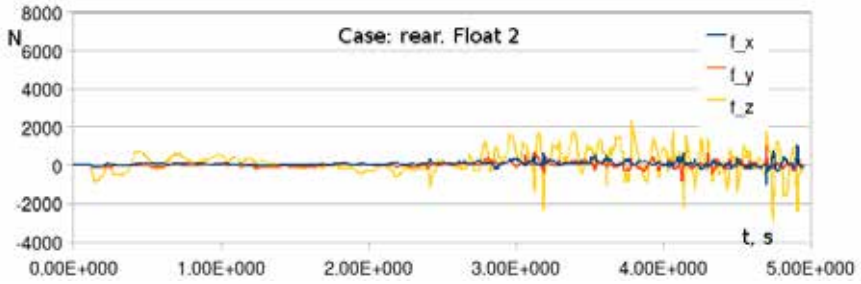
temporal evolution of the forces acting on the first and the second floats respectively (again, due to the symmetry, results for float N3 and N4 are equivalent to those of N1 and N2 respectively, see Fig. 2.9). In the case of accumulation of water at the rear part of the vehicle, only rear floats (N1 and N3) are heavily loaded(see Fig. 2.29). The maximum force in vertical direction of value $\mathbf{f}_z$=4.2 [kN] is exerted upon the rear float at ≈1.5 [s]. Maximum forces in x and y directions amount to 1.8 [kN] and 0.5 [kN] respectively. Forces exerted upon Floats 1 and 3 reach at most 0.5 [kN] and are negligible.

Case lat:

- Mass $m = 6118.5[kg]$



(a) Float 1



(b) Float 2

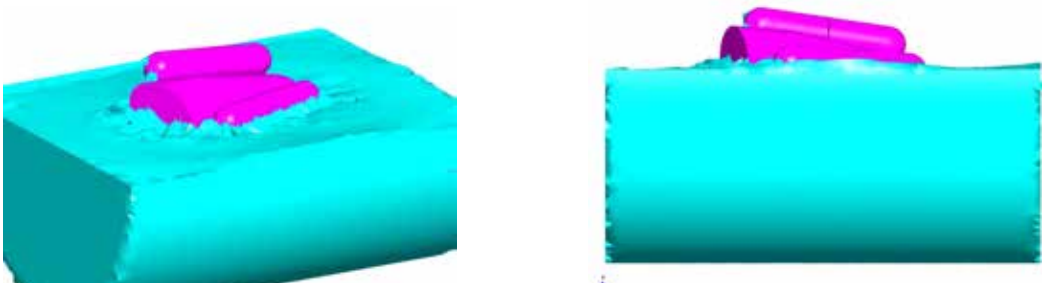
Figure 2.29: Temporal evolution of forces acting on the floats under assumption of water accumulation at the rear

- Center of gravity: (1.72, -0.17, -0.09)
- Inertia tensor: I=

Ixx	Ixy	Ixz	=	2066	-268	23
Iyx	Iyy	Iyz		-268	9401	60
Izx	Izy	Izz		23	60	9922

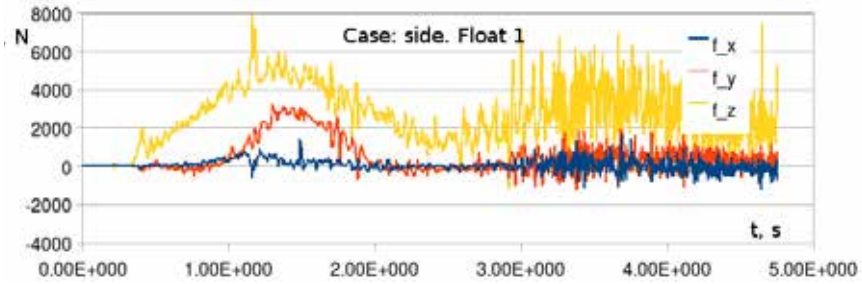
The equilibrium position obtained in this case is depicted in Fig. 2.30.

Fig. 2.31 represents temporal evolution acting on the first and the second floats respectively. Floats N3 and N4 in this case undergo much lighter loadings and thus are of no interest in the analysis. One can see in Fig. 2.31, that under an assumption of water accumulation at the side of

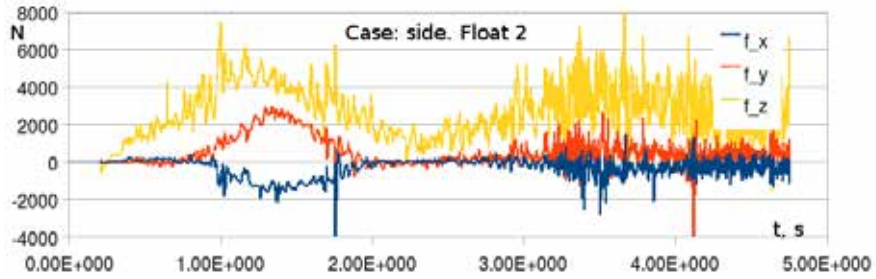


(a) 3D view (b) XZ-plane

Figure 2.30: Equilibrium position of the capsule under the assumption of lateral water accumulation



(a) Float 1



(b) Float 2

Figure 2.31: Temporal evolution of forces acting on the floats under assumption of lateral water accumulation

the vehicle both Float1 and Float2 undergo considerable loading. The maximum force in vertical direction of value $\mathbf{f}_z \approx 5$ [kN] is exerted upon floats 1 and 2 at ≈ 1.5 [s]. Maximal forces in x and y directions amount to 3 [kN] and 0.5 [kN] for the rear float (Float1); 3 [kN] and 1.5 [kN] (Float2), respectively. Forces exerted upon Floats 1 and 3 are negligible.

In all the analyzed cases there exist no danger of sinking and no tendency to flip-over was found.

2.7 Summary and conclusions

In this chapter we have introduced the basic features of the classical Particle Finite Element Method. The advantages of Lagrangian methods for solution of the free-surface flows simulation were emphasized. Equations of motion of a viscous incompressible fluid in the Lagrangian framework were presented. Pressure stabilization based upon the FIC technique was outlined. The re-meshing tools necessary when a Lagrangian description is used for domains undergoing large deformations were introduced. Additional ad-hoc techniques ensuring mesh quality were described.

Next, the governing equations of the rigid body mechanics were presented. Finally the PFEM and the rigid-body model were used to define a partitioned strategy for the simulation of a challenging industrial FSI problem, namely the sea-landing of a satellite capsule. The Dirichlet-Neumann

coupling strategy was used, enriched by introducing the interface FSI Laplacian accounting for the motion of the structure in calculation of pressure at the second stage of the fractional step scheme, and thus capturing the local pressure wave at the fluid-structure interface.

The impact analysis has shown maximum impact accelerations lying within an expected range according to the experimental measurements performed by an industrial project partner.

The simulations performed demonstrated that the existing float design ensures that the vehicle remains on the water surface. The assumption of the most unfavorable case, namely complete filling of the space between the external panel and the internal housing with water leads to the maximum load exerted upon the floats of ≈ 6 [kN]. This happens in case of water accumulation at the front part of the vehicle. In either of the three assumptions of water distribution (front, rear, lateral) only two floats out of 4 actually undergo considerable loading.

It was observed in all the simulated cases, that none of them is flip-over prone. That is, the floats assure not only floatability, but also stability of the capsule.

Chapter 3

Quasi-incompressible Lagrangian fluid and monolithic FSI

3.1 Introduction and outline

The previous chapter served for introducing the Lagrangian FEM concepts and presented the classical Particle Finite Element Method. The PFEM formulation presented was used as one of the ingredients of the partitioned Lagrangian-Lagrangian fluid-structure interaction solver. The current chapter defines the core development of the present work. It explores the maximum benefit that can be taken using a Lagrangian fluid formulation for the fluid-structure interaction. We propose an innovative quasi-incompressible fluid formulation and on its basis derive a monolithic FSI strategy. The developments of this chapter are motivated by FSI problems involving light-weight structures, an area where most of the partitioned methods fail.

Here we propose a monolithic method, where the linearized global system (containing both fluid and solid sub-domains) is solved exclusively for displacements. Pressure condensation is achieved by introducing a slight compressibility for the fluid. Once the variable heterogeneity is removed, the system can be solved by a simple diagonally preconditioned Conjugate Gradient method to give feasible convergence rates. Symmetry of the global discrete system results from the Lagrangian description of both sub-domains. We adopt in the following the philosophy of the PFEM introduced in **Chapter 2** with respect to free-surface identification, variable storage and re-meshing.

As we shall see, one of the advantages of using the Lagrangian quasi-incompressible fluid formulation is that it enables the definition of a monolithic FS system in conjunction with a vast majority

of existing structural codes(solids, shells, membranes) without any modifications in those. This ensures the re-usability of the existing structural codes, which is a considerable gain for a monolithic method. The monolithic system in the proposed method is assembled in a way very similar to that for a single filed problem, namely by looping over all elements, where in the Newton-Raphson loop each element contributes its dynamic residual and the dynamic tangent stiffness to the discrete system matrix. The only term that requires additional care for an efficient implementation is the one corresponding to the volumetric stress in the fluid. The fluid-structure interaction at the interface is an intrinsic feature of the method in the sense that using a single displacement variable across both the fuild and structure domains eliminates the need for any special treatment for enforcing consistent interface conditions.

The chapter is organized as follows. An updated Lagrangian quasi-incompressible fluid formulation is derived first. The global pressure condensation associated with the linear pressure interpolation is then described. It is shown that the computational effort introduced by the global pressure condensation is minimized if a matrix-free iterative method is used for the solution of the linearized system.

Next the monolithic FSI coupling strategy is presented. No interface equation needs to be specified as the interaction is an intrinsic feature of the monolithic method when the fluid and the solid share the same unknowns and the FS domain possesses unique discretization. It is emphasized that the proposed fluid can be used in conjunction with any Lagrangian displacement based structure to naturally define a monolithic FS system. The effect of the bulk modulus value upon the convergence of non-linear loop and linear solvers is addressed. The chapter concludes with numerical examples showing the functionality of the method and highlighting the application area where it is of particular advantage.

3.2 Motivation for development of a monolithic Lagrangian FSI method

Two main ingredients that define any FSI strategy are the kinematic description of the sub-domains and the algorithm for their coupling. We start by reviewing existing possibilities and motivate the choice made here.

Partitioned Vs. Monolithic The most common way of approaching fluid-structure interaction problem relies on the domain decomposition and separate solution of the involved subsystems (the

fluid and the structure) equipped with the coupling iterations. The splitting of the coupled problem into fluid and structure sub-problems permits to re-use the existing codes best suitable for each sub-domain. In the introduction to the previous chapter we mentioned the immersed boundary technique that basically consist in keeping standard Lagrangian approach for the structure but solving the fluid on the fixed grid (Eulerian). Another class of methods relies on defining a partitioned strategy keeping Lagrangian solid description and coupling it to a Lagrangian or ALE-based fluid. One such approach was presented in **Chapter 2**, where a mesh-moving fluid method was equipped with a re-meshing tool, facilitating the treatment of large domain deformations.

In any case when partitioned approaches are chosen, coupling iterations over the fluid-structure interface must be performed. In the previous chapter we have introduced the Dirichlet-Neumann coupling. If the surface meshes of the sub-domains are fitting (which can be easily obtained in ALE or Lagrangian approaches), the application of the boundary conditions is straight-forward: they are applied at the nodes of the interface elements. If the immersed boundary methodology is chosen, the interface conditions have to be enforced in a weak way by testing it with the test function along the interface.

Initially introduced in the field of aeroelasticity, Dirichlet-Neumann algorithms give optimal convergence rates (usually coupling iterations are done in terms of interface displacements) for the applications where the fluid density is much smaller than that of the structure (e.g. air and steel). However for light-weight structures and in particular as the density ratio between the involved fluid and solid is close to one ($\frac{\rho_f}{\rho_s} \approx 1$) the convergence drastically deteriorates. An intuitive insight of the cause can be gained from the following explanation: if the structural behavior is close to rigid, structural part in a way dominates the interaction, making a correct prediction of the interface displacements. However, if the structure is light and flexible, the fluid forces have a great impact upon the displacement of the structure, making the predictor's guess with regard to the interface displacement far from the real value.

Usual remedies for circumventing these difficulties are the relaxation schemes that use the information about the interface residuals at the preceding step for prediction of the interface displacement (e.g. Aitken or steepest descent) [48]. We should also mention another possibility that is based on using different, namely Robin-Robin[1] or Robin-Dirichlet boundary conditions, instead of Dirichlet-Neumann ones [49], [50].

A qualitatively different approach consists in designing a monolithic method. The coupling

[1]Robin boundary conditions are a weighed combination of Dirichlet boundary conditions and Neumann boundary conditions.

iterations that are the **intrinsic features of partitioned methods** simply do not exist there. This chapter aims at presenting such a methodology and introduces an approach for treating the strongly coupled fluid-structure systems involving light-weight structures in a monolithic way. By a monolithic FSI method we mean that a unique discrete system of equations is written, taking into account both the sub-domains (the fluid and the structure)[2]. The non-linear FSI system is solved as a whole iteratively (usually using Newton's method). Monolithic methods are robust as they do not introduce any domain splitting error. However they lead to large and generally poorly conditioned systems due to the different scaling of variables (velocities, displacements, pressure) entering the multi-field problem. Therefore the efficiency of monolithic methods for solution of large-scale problems is strongly dependent on the availability of sophisticated preconditioners, as direct solvers (less sensitive to system conditioning) cannot be used for cases involving millions of degrees of freedom. Whereas the problem of partitioned methods is the convergence of the interface coupling, the problem of monolithic methods is the convergence of the iterative solver applied to the solution of the linearized system, when the problem size precludes using direct solvers.

We have chosen a Lagrangian description for the fluid and the solid for a monolithic FSI strategy in order to explore the maximum advantage one can obtain by using the same formulations for both the sub-domains and the resulting symmetry of the monolithic FSI system. We shall see in detail what can be benefited from this choice.

Finally, we would like to mention that there exists also a possibility of defining a monolithic method describing both the fluid and the structure in the Eulerian framework. The interface is described in this case implicitly, usually by level set functions. The main drawback, however, originates from the fact that the kinematics of the vast majority of existing structural formulations is described using the Lagrangian framework, which is natural for solids. So from the point of view of the code re-usability this option is highly unfavorable. Additionally for Eulerian solids the constitutive laws (except for linear elasticity, where certain know-how with respect to formulation in the Eulerian framework exists [51]) are cumbersome and in many cases not available. A monolithic Eulerian-Eulerian FSI scheme is presented and used in [52].

Monolithic FSI When designing a monolithic FSI method the following points have to be considered

- The resulting linearized system should be solvable by conventional iterative methods (adequate

[2]Not to be confused with the monolithic single-field fluid, which implies simultaneous solution for velocity and pressure, in contrast with fractional techniques

system conditioning)

- The formulation must ensure good convergence rates of the non-linear iterations procedure (linearization)
- The code re-usability: for a FSI monolithic solver possibility to utilize the structural codes without any modifications in those would be of a great advantage
- Natural coupling: the system assembly and solve should be as similar to that of a single field problem as possible.

Unfortunately it is impossible to satisfy all the above at once to the same degree. If no special care is taken, conventional iterative linear solvers applied to solution of a "heterogeneous"[3] monolithic system either converge extremely slow or do not work at all. For example, in the immersed boundary method applied as it is in the monolithic context, the system resulting system conditioning is extremely poor due to the presence of four types of variables, which are pressure and velocities for the fluid, displacements for the structure and the Lagrange multiplier appearing in the interface equation [40].

Theoretically, there exists a possibility to define the monolithic system involving a standard Lagrangian structure and the Lagrangian PFEM-based incompressible fluid, as described in **Chapter 2**. In comparison with the Eulerian fluid-Lagrangian structure approach the gain is the natural definition of the fluid-structure interface and absence of the Lagrange multiplier variable. However different variable scaling presence of pressure/velocity and displacement is enough to keep the system conditioning poor[4].

To eliminate this drawback, Idelsohn et al. in [21] proposed to treat the fluid as slightly compressible, using condensation of the pressure variables at the elemental level in conjunction with discontinuous pressure elements. Additionally, switching from a displacement-based to velocity-based formulation for the solid proposed in [21], facilitated treating the coupled system in a completely unified manner, in the sense of the unique degrees of freedom shared by the elements of fluid in contact with structure. The resulting formulation is purely velocity based and has proved to be efficient for definition of the monolithic system in the sense of system conditioning and thus a quick convergence of the iterative solvers applied to the linearized system. However, constant pressure

[3] by "heterogeneous" we mean the presence of different variables in the system, e.g. pressure, velocities and displacements, typical for FSI problems

[4] The bad conditioning due to variable heterogeneity is present also in a purely fluid problem. Indeed complications in solving for pressure and velocity at once due to poor system conditioning gave rise to the popularity of fractional step methods. Decoupling the solution for the pressure and the velocity results in much better conditioned subsystems. Of course the conditioning problem only amplifies in FSI, since the equilibrium equations for the structure (written in terms of displacements) enters the monolithic system.

elements suffered from volumetric locking already for moderate values of the bulk modulus, thus making application in the incompressible limit problematic. On the other hand, small values of the bulk modulus also might lead to considerable volume losses. Importantly, the value of the bulk modulus for which locking behavior appears depends also on the mesh size. The locking phenomenon encountered in constant pressure triangles is well known in the area of incompressible elastic in solids [53].

The idea followed here is to derive a fluid formulation based on a continuous displacement-pressure interpolation pair. We choose displacements (rather than velocities) as primary variable for the fluid for the purpose of the coupling with most of existing structural codes, that are generally displacement-based. Continuous pressure interpolation leads to the diminishing of the number of constraints imposed upon the displacement field and (if pressure stability is assured) does not lead to locking phenomenon for any value of bulk modulus. Two problems arise however when continuous pressure is used. First one is the impossibility of elemental pressure condensation. Condensation is possible only at the global level, which at a first glance would require matrix inversions, that is a computationally expensive procedure. Second problem originates from the instability of the P1/Q1 element in the incompressible limit, which is a qualitative problem. The conventional pressure stabilization techniques cannot be used as they would require inversion of the global pressure Laplacian in case pressure is condensed. A different methodology proposed here is based upon the difference of the consistent and lumped mass matrices multiplying the pressure in the continuity equation of the quasi-incompressible fluid. As we shall see, this permits on one hand to stabilize the pressure and on the other hand prevent matrix inversion as the unknown pressure term shall be multiplied with the lumped mass matrix. Once pressure is condensed, the solution for the displacement of the entire FSI system is decoupled from the solution for the pressure. In fact in the pressure step no system is solved. The entire FSI strategy at one time step can be summarized as follows:

1. Start the non-linear loop
    a. Solve the monolithic FSI system for displacements
    b. Compute the fluid pressure from the displacement field
    c. Add the gradient of the obtained pressure to the residual of the momentum equation
    d. Update position of the nodes
    e. Go to 1.(a) until the convergence in terms of displacements is achieved
2. End non-linear loop, re-mesh

In the following section we will develop in detail this formulation and derive the discrete equations first for the fluid and then for the entire FSI system.

3.3 Updated Lagrangian FE formulation for the quasi-incompressible fluid

In this section we derive the formulation for the fluid. Even though the section is devoted to the fluid equations, we shall keep in mind that some of the methodologies and approximations (e.g. slight compressibility, non-conventional pressure stabilization) are introduced specifically for FSI purposes. The fluid formulation proposed next is not claimed to be an alternative to the standard incompressible fluid for single-field problems. It can also be used in that context, even though most of its advantages are exhibited in the coupled context. The strategies for the free-surface identification, variable storage and re-meshing are completely inherited from the classical PFEM approach introduced in the previous chapter and will not be repeated here. The difference consists in the governing equations and the way of solving them.

Quasi-incompressibility The term "quasi-incompressibility" that we shall use must be clarified first. 'Quasi-incompressibility" implies that the density changes are small enough to neglect the coupling of the momentum and continuity equations with the energy equation. On the other hand, it means that the continuity equation for the incompressible fluid ($\nabla \cdot \mathbf{v} = 0$) is not satisfied exactly. Instead it is approximated by the compressible version

$$\frac{1}{\kappa}\frac{\partial p}{\partial t} = \nabla \cdot \mathbf{v} \tag{3.1}$$

with κ being the fluid compressibility constant (or bulk modulus). The advantage of the compressible continuity equation is the possibility to relate the pressure with the velocity field. As we shall see further it is of crucial importance for the linearization of the discrete system proposed here.

Obviously if the bulk modulus tends to infinity, the incompressible version of the continuity equation is recovered. From the abstract view point, κ can be viewed as a penalty parameter. Indeed, the quasi-incompressible equations can be obtained by penalizing the incompressibility condition, see [54].

It is important to clearly understand the validity of applying the quasi-incompressible equations for modeling the incompressible fluid. This shall shed some light upon the values of the bulk modulus

that should be used in modeling[5].

The very first consideration is to match the initial assumption, that the volume change encountered in the simulation should be sufficiently small. As we shall see further on, this is satisfied already for moderate values of bulk modulus.

Second consideration imposes stricter limits upon the minimum value of the bulk modulus. In the truly incompressible fluid, the speed of sound propagation is infinite[6]. In the compressible (or quasi-incompressible) fluid the sound speed propagation and the compressibility are interconnected by

$$\kappa = -\rho c^2 \tag{3.2}$$

From here one can see that for correct prediction of the physical phenomenon the velocities encountered in the modeled problem should be several orders of magnitude smaller than the finite sound speed introduced by the compressibility. This can be also understood in a different way in the context of discrete problems: the characteristic length of the problem should be much smaller than the time step multiplied by the sound speed. The utilization of high values of bulk modulus is particularly important in the simulation of the impacts, where the velocities at the short moment of impact are very high.

At this point we introduce the governing equations for a quasi-incompressible Newtonian fluid written in the Lagrangian reference frame. The momentum equation (written taking the current (unknown) configuration Ω_t as the reference configuration) is

$$\rho \frac{\partial \mathbf{v}}{\partial t} - \mu \nabla \cdot \nabla^S \mathbf{v} + \nabla p = \rho \mathbf{f} \tag{3.3}$$

in Ω for $t \in (0, T)$, where $\mathbf{v}$ is the velocity field, μ the dynamic viscosity, $\mathbf{f}$ the body force vector, p is the pressure and ∇^S the symmetric gradient operator. Next we neglect the $\mu \nabla (\nabla \cdot \mathbf{v})$ term, assuming that both the product of fluid viscosity with velocity divergence is small. In order to obtain the equations written in terms of the same variables as structure, we switch to displacements, defining $\mathbf{u} = \mathbf{x} - \mathbf{x}_0$. This gives

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} - \mu \Delta \frac{\partial \mathbf{u}}{\partial t} + \nabla p = \rho \mathbf{f} \tag{3.4}$$

[5] In fact, water which is the main objective of current work is a compressible fluid with the bulk modulus $\kappa \approx 2E09[Pa]$. However, as we shall see further, using the values of bulk modulus as high as the physical one drastically deteriorates the efficiency of the proposed formulation.

[6] This is true on the continuous level and for the monolithic fluid formulations. The fractional step formulations introduce finite propagation speeds.

The continuity equation in the Lagrangian framework reads

$$\frac{\partial \rho}{\partial t} + \rho \nabla \cdot \mathbf{v} = 0 \tag{3.5}$$

For isothermal flows

$$\frac{\partial \rho}{\partial t} = -\frac{1}{c^2}\frac{\partial p}{\partial t} \tag{3.6}$$

where c is the velocity of sound propagation in the medium. Using 3.6 and substituting velocity by the derivative of the displacement, we obtain the constitutive equation for the pressure as

$$\frac{\partial p}{\partial t} = -\rho c^2 \nabla \cdot \frac{\partial \mathbf{u}}{\partial t} \tag{3.7}$$

Substituting $\kappa = \rho c^2$ Eq. (3.7) can be written as

$$\frac{\partial p}{\partial t} = -\kappa \nabla \cdot \frac{\partial \mathbf{u}}{\partial t} \tag{3.8}$$

On the other hand by definition one can relate directly the bulk modulus to the variation of a given control volume V.

$$\kappa \equiv -\frac{\partial p}{\partial V} V \tag{3.9}$$

where V is the volume. This equation permits the computation of the pressure from the initial and final volumes. The use of such equation is not really appealing at the continuum level, as the definition of "control volume" is somewhat arbitrary. As we will see later however its discrete counterpart will present some important advantages, which will be exploited in the strategy we propose.

The equations that define the fluid motion are Eq. (3.4), Eq. (3.8) and Eq. (3.9). At the continuous level Eqs. (3.8) and (3.9) are equivalent. Eq. (3.8) relates the temporal derivative of pressure with the gradient of the velocity field, whereas Eq. (3.9) expresses the pressure in terms of the volume change without knowing the temporal dependence but knowing the current configuration. Both equations will be used in the current work: Eq. (3.8) for relating pressure and displacement, and Eq. (3.9) for pressure recovery, once the new configuration is found.

Finally we can write:

$$\delta p = -\kappa \frac{\delta V}{V} = \int_t -\kappa \nabla \cdot \frac{\partial \mathbf{u}}{\partial t} dt \tag{3.10}$$

where δp and δV are finite pressure and volume increments, respectively.

3.3.1 Discretization

The weak form of Eq. (3.3) and Eq. (3.8) using a linear displacement-pressure interpolation reads

$$\left(\rho\frac{\partial^2\mathbf{u}}{\partial t^2},\mathbf{w}\right)+\mu\left(\nabla\frac{\partial\mathbf{u}}{\partial t},\nabla\mathbf{w}\right)+(\nabla p,\mathbf{w})=\rho\langle\mathbf{f},\mathbf{w}\rangle \tag{3.11}$$

$$\left(q,\frac{\partial p}{\partial t}\right)=-\kappa\left(q,\nabla\cdot\frac{\partial\mathbf{u}}{\partial t}\right) \tag{3.12}$$

where $\mathbf{w}$ and q are the displacement and pressure test functions. In the equations above $(\cdot,\mathbf{w})$ and $\langle\cdot,\mathbf{w}\rangle$ are the standard bilinear and linear forms in $\mathbf{w}$ and $(\cdot,q)$ is the bilinear form in q. It is important to keep in mind that the choice of an updated Lagrangian description means that all the spatial integrals are evaluated in the current configuration Ω_t. The semi-discrete equations then read

$$\rho\mathbf{M}\frac{\partial^2\bar{\mathbf{u}}}{\partial t^2}-\mu\mathbf{L}\frac{\partial\bar{\mathbf{u}}}{\partial t}+\mathbf{G}\bar{p}=\mathbf{F} \tag{3.13}$$

$$\mathbf{M}_p\frac{\partial\bar{p}}{\partial t}=-\kappa\mathbf{D}\frac{\partial\bar{\mathbf{u}}}{\partial t} \tag{3.14}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{M}_p$ - the pressure mass matrix both used in the lumped format (note that if in the mass matrix super-index "c" is omitted, we assume that it is taken in the lumped format), $\mathbf{L}$ the Laplacian, $\mathbf{G}$ the gradient and $\mathbf{D}$ the divergence matrices. These follow the definitions introduced in **Chapter 2**.

Eq. (3.13) and Eq. (3.14) are still continuous in time. The relation between the displacement and its derivatives will be thus provided once a time integration scheme is chosen as it is usually done for the solids.

At thus point we perform the time discretization and express the momentum and mass conservation equations in residual form in order to perform next the linearization. For the sake of clarity we shall illustrate the method using the simplest implicit time integration scheme, namely the Backward Euler, while in the actual implementation the Newmark-Bossak scheme was used. The momentum residual equation integrated in time using Backward Euler scheme gives

$$\bar{\mathbf{r}}_m:=\mathbf{F}_{n+1}-\mathbf{G}\bar{p}_{n+1}+\mu\mathbf{L}\frac{\bar{\mathbf{u}}_{n+1}-\bar{\mathbf{u}}_n}{\Delta t}-\rho\mathbf{M}\frac{\bar{\mathbf{u}}_{n+1}-2\bar{\mathbf{u}}_n+\bar{\mathbf{u}}_{n-1}}{\Delta t^2} \tag{3.15}$$

Note, that all the discrete operators in Eq. (3.15) are the ones corresponding to the unknown current configuration Ω_{n+1}[7].

[7]It was possible also to choose $\Omega(t_n)$ as the reference configuration. This would imply two major changes. First one, is that the associated stress/strain measures are Piola-Kirchhoff stresses and Green-Lagrange strains instead of

For the pressure, on the other hand, the choice of the time integration scheme is not so straightforward. Two choices exist indeed to relate the pressure and the variation of the displacement field. The first one comes from Eq. (3.8), relating the pressure and the velocity divergence as:

$$\mathbf{M}_p \delta\bar{p} = -\kappa \int_{t_n}^{t_{n+1}} \mathbf{D}(t)\bar{\mathbf{v}} dt \tag{3.16}$$

The difficulty arises from the fact that the discrete divergence operator varies in time, thus making difficult the exact computation of the integral. The simplest possibility to "avoid" this difficulty is to use an approximation, i.e.

$$\mathbf{M}_p \delta\bar{p} = -\kappa \int_{t_n}^{t_{n+1}} \mathbf{D}(t)\bar{\mathbf{v}} dt \approx -\kappa \mathbf{D} \int_{t_n}^{t_{n+1}} \bar{\mathbf{v}} dt \approx -\kappa \mathbf{D}_{n+1} \Delta\bar{\mathbf{u}}_{n+1} \tag{3.17}$$

This approach is indeed possible, nevertheless it introduces an additional error due to the assumption of the time independence of the divergence operator $\mathbf{D}$, an error that becomes especially considerable for large time steps and large rotations.

The second option does not involve any approximation. The idea consists in using Eq. (3.9), which permits the computation of the pressure directly from a given discrete volume change, just knowing the initial and final volumes and without knowing "what happens in-between"[8]. To use this formula in the discrete setting we shall define the "control volume" next. In the following we will associate to each node of the FE mesh a nodal volume defined in such a way that it coincides with the diagonal entry of the diagonalized pressure mass matrix.

$$\bar{V}_I := \mathbf{M}_{p,II} \tag{3.18}$$

Thus the nodal pressure increment can be *exactly* related to the nodal volume variation by the formula

$$\bar{p}_{n+1,J} = \bar{p}_{n,J} - \kappa \frac{\bar{V}_{n+1,J} - \bar{V}_{n,J}}{\bar{V}_{n,J}} \tag{3.19}$$

Cauchy stresses and Almansi strains used here. At the discrete level the change would affect the discrete operators. The derivation of the shape functions with respect to the current position (necessary e.g. for the discrete gradient, divergence and Laplacian matrices) would involve deformation gradients. This is omitted if the integrals are taken in $\Omega(t_{n+1})$. However this implies that all the geometrical data (position, shape functions etc) must be recomputed at every non-linear iteration step.

[8] A similar way of relating the pressure to the volume change can be found in [55].

or in the context of Newtons' iterations loop as:

$$\bar{p}_{n+1,J}^{i+1} = \bar{p}_{n,J} - \kappa \frac{\bar{V}_{n+1,J}^{i+1} - \bar{V}_{n,J}}{\bar{V}_{n,J}} \tag{3.20}$$

where i stands for the non-linear iteration index.

In this sense Eq. (3.20) is clearly superior to Eq. (3.17) as it does not involve any additional approximation.

Remark: It is interesting to observe that using Eq. (3.20) can be interpreted as imposing mass conservation at a number of integration points (the nodes of the FE mesh) which is sensibly smaller than the number of Gauss points used in the traditional element based approach. Such reduction of the total number of volumetric constraints provides an heuristic justification of the fact that the proposed method is locking-free. This argument was used for example in [56] to justify the F-bar method for low order elements.

The pressure residual is defined as

$$\bar{r}_p := \bar{p}_{n,J} - \bar{p}_{n+1,J}^{i+1} - \kappa \frac{\bar{V}_{n+1,J}^{i+1} - \bar{V}_{n,J}}{\bar{V}_{n+1,J}^{i+1}} \tag{3.21}$$

which can be written equivalently

$$\bar{r}_p := \mathbf{M}_{p,JJ}\bar{p}_{n,J} - \mathbf{M}_{p,JJ}\bar{p}_{n+1,J}^{i+1} - \kappa \left(\bar{V}_{n+1,J}^{i+1} - \bar{V}_{n,J} \right) \tag{3.22}$$

For convenience, we finally introduce the symbol

$$\bar{\mathbf{r}} := \begin{pmatrix} \bar{\mathbf{r}}_m \\ \bar{r}_p \end{pmatrix} = 0 \tag{3.23}$$

to denote the set of residual equations.

3.3.2 Linearization

(3.23) is an equation system, which is non-linear because all the operators are written at the unknown configuration Ω_{n+1}. For solution of this system Newton's method will be applied. In addition to the already defined dynamic residual, the tangent matrix $\mathbf{H}$ needs to be defined.

By definition the tangent stiffness is the derivative of the residual with respect to the primary

variables

$$\mathbf{H} = -\frac{\partial \bar{\mathbf{r}}\left(\bar{\mathbf{u}}^i, \bar{p}^i\right)}{\partial \bar{\mathbf{u}}, \bar{p}} \tag{3.24}$$

The Newton method can be summarized as follows:

1. solve $\mathbf{H}\begin{pmatrix} d\bar{\mathbf{u}} \\ d\bar{p} \end{pmatrix} = \bar{\mathbf{r}}\left(\bar{\mathbf{u}}^i, \bar{p}^i\right)$ for $d\bar{\mathbf{u}}$ and $d\bar{p}$
2. update $\bar{\mathbf{u}}^{i+1} = \bar{\mathbf{u}}^i + d\bar{\mathbf{u}}^i$ and $\bar{p}^{i+1} = \bar{p}^i + d\bar{p}^i$
3. Go to 1 until convergence in $\bar{\mathbf{u}}$

where $d\bar{\mathbf{u}}$ and $d\bar{p}$ are the displacement and pressure increments, n stands for the time step and i is a non-linear iteration index.

The Lagrangian description in contrast to the Eulerian theoretically implies that all the discrete operators entering the problem are non-linear. Despite of being obvious it might be a point of confusion, since in the classical fluid formulations (Eulerian) the non-linearity originates entirely from the convective term, due to dependence of the convective operator upon the unknown velocity. The rest of the operators remain constant in the Eulerian description. In turn, in Lagrangian formulations the "troublesome" convective term does not exist, the problem however shifts to the presence of the operators changing according to the mesh position.

Before deriving the expression of the tangent stiffness matrix several observations must be made:

- Re-meshing is performed after every time step (following the basic PFEM formulation as described in **Chapter 2**) and not at every non-linear iteration, which leads to the following consequences

  a) The nodal connectivities do not alter within one time step

  b) The mass matrix and external force vectors remain unchanged (they do not contain derivatives of the shape functions)

  c) The mass matrix and external force vectors must be updated at each time step only, but not within the iterative loop

The non-linearity that needs to be considered when deriving the dynamic tangent stiffness is thus contained only in the internal force terms, that are the viscous term $\mu\mathbf{L}\bar{\mathbf{v}}_{n+1}$ and the volumetric term $\mathbf{G}\bar{p}_{n+1}$. The discrete Laplacian and the gradient operators contain derivatives of the shape functions with respect to unknown current configuration $\bar{\mathbf{x}}_{n+1}$. We assume that the discrete operators $\mathbf{L}$ and $\mathbf{G}$ do not change within a non-linear iteration. However this does not imply that they do not change

within the time step. At every non-linear iteration $\mathbf{L}$ and $\mathbf{G}$ are recomputed in the newly obtained configuration and then used in the next iteration. So the continuously varying in time $\mathbf{L}(t)$ and $\mathbf{G}(t)$ are approximated within a time step by a series of constant operators $(\mathbf{L}^0, \mathbf{L}^1 ... \mathbf{L}^i)$ and $(\mathbf{G}^0, \mathbf{G}^1 ... \mathbf{G}^i)$. This classifies the linearization method as "inexact-Newton". The exact linearization of the gradient and Laplacian operators would be cumbersome and noticeably decrease the computational efficiency of the entire algorithm.

The linearization of the pressure residual (Eq. (3.22))gives:

$$\frac{\partial \bar{r}_p}{\partial \bar{p}} = \mathbf{M}_p \tag{3.25}$$

and

$$\frac{\partial \bar{r}_p}{\partial \bar{\mathbf{u}}} = \kappa \frac{\partial \left(\bar{V}^{i+1}_{n+1,J} - \bar{V}_{n,J} \right)}{\partial \bar{\mathbf{u}}} = \kappa \mathbf{D} \tag{3.26}$$

Linearization of the momentum residual results in:

$$\frac{\partial \bar{\mathbf{r}}_m}{\partial \bar{\mathbf{u}}} = \rho \frac{\mathbf{M}}{\Delta t^2} - \mu \frac{\mathbf{L}}{\Delta t} \tag{3.27}$$

and

$$\frac{\partial \bar{\mathbf{r}}_m}{\partial \bar{p}} = \mathbf{G} \tag{3.28}$$

This allows to write the Newton system as

$$\begin{pmatrix} \rho \frac{\mathbf{M}}{\Delta t^2} - \mu \frac{\mathbf{L}}{\Delta t} & \mathbf{G} \\ \mathbf{D} & \frac{1}{\kappa} \mathbf{M}_p \end{pmatrix} \begin{pmatrix} d\bar{\mathbf{u}} \\ d\bar{p} \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{r}}_m \left(\bar{\mathbf{u}}^i_{n+1}, \bar{p}^i_{n+1} \right) \\ \frac{1}{\kappa} \bar{r}_p \left(\bar{\mathbf{u}}^i_{n+1}, \bar{p}^i_{n+1} \right) \end{pmatrix} \tag{3.29}$$

The system described in Eq. (3.29) is known as a "monolithic fluid" system (NB: not to be confused with the "monolithic FSI" system.). The system is classified as "strongly coupled" in the sense that the solution for both displacement and pressure has to be performed simultaneously. Comparing the form of the resulting system with the form emerging in the truly incompressible case, one can notice a mass matrix appearing in the lower diagonal position of the system, where 0 appears in the incompressible case. In principle, the presence of the non-zero entity already preserves to some extent the stability of the numerical solution. However, as we shall see, once the incompressible limit is approached, i.e. the bulk modulus increases, the pressure stability becomes problematic. The next section is be devoted to the discussion of the behavior of the proposed method at high values of bulk modulus. Some problems will be identified in this context, and a modification of the

"constitutive" pressure equation will be introduced to improve the stability properties.

3.3.3 Pressure stability

Before addressing the pressure stabilization technique we propose in this work, we start with a brief excourse in the history, highlighting the origins of the pressure instability. To gain the insight it is sufficient to consider the stationary Stokes problem:

$$\nabla p - \mu \nabla \mathbf{v} = \mathbf{f} \tag{3.30}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{3.31}$$

leading to the following discrete system

$$\begin{pmatrix} -\mu\mathbf{L} & \mathbf{G} \\ \mathbf{D} & 0 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{v}} \\ \bar{p} \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ 0 \end{pmatrix} \tag{3.32}$$

where $\mathbf{L}$ is the Laplacian matrix, $\mathbf{D}$ and $\mathbf{G}$ are the divergence and gradient matrices and $\mathbf{F}$ is the body force vector term. The structure of the discrete system is peculiar due to the presence of null-matrix in the lower diagonal. It can be shown that this algebraic system can only be solved provided that the kernel of the gradient matrix $\mathbf{G}$ is zero. This leads to the solvability of the global system and thus to the unique solution for the velocity and pressure.

To understand better what influence the kernel of the gradient operator has upon the solvability of the system, let us assume that the system is solved in a partitioned manner. This is done by obtaining from the first equation $\bar{\mathbf{v}} = \mathbf{K}^{-1}(\mathbf{F} - \mathbf{G}\bar{p})$ and plugging it into the second equation resulting in the so-called "pressure shur equation" :

$$(-\mathbf{D}\mathbf{L}^{-1}\mathbf{G})\bar{p} = \mu \mathbf{D}\mathbf{K}^{-1}\mathbf{F} \tag{3.33}$$

The Eq. (3.33) can be solved if the matrix $\mathbf{D}\mathbf{K}^{-1}\mathbf{G}$ is positive definite (which assures its invertibility). This is satisfied if and only if the discrete gradient $\mathbf{G}$ has zero kernel [9].

It turns out that this is assured only if the velocity-pressure interpolation pairs are taken from the velocity and pressure spaces ($\mathcal{V}^h$ and $\mathcal{Q}^h$) that respect the following inf-sup condition:

$$inf_{q^h \in \mathcal{Q}^h} \, sup_{w^h \in \mathcal{V}^h} \frac{(q^h, \nabla \cdot w^h)}{||q||_0 ||w^h||_1} > \alpha > 0 \tag{3.34}$$

where α is a positive constant independent of the mesh size h. This inf-sup condition is known as Ladyzhenskaya-Babushka-Brezzi (LBB) condition [9].

Unfortunately the most widely used linear velocity-pressure interpolation pairs do not satisfy the LBB condition [57]. Using LBB stable elements, such as Q2Q1 (Taylor-Hood element, based upon continuous quadratic velocity, continuous bilinear pressure) or mini element (continuous linear velocity+bubble function, continuous linear pressure) is less favorable due to the complexity of the associated implementation and clearly lower computational efficiency than that of simplicial elements. As it is well known, there exist possibilities for circumventing the LBB condition, thus permitting the use of the velocity-pressure pairs that are unstable when standard Galerkin formulations are used. The basic idea behind pressure stabilization methods is to modify the weak form of the problem in such a way that the positive definiteness of the matrix $\begin{pmatrix} -\mathbf{L} & \mathbf{G} \\ \mathbf{D} & 0 \end{pmatrix}$ is enforced. Most of the well-established stabilization techniques (e.g. Galerkin/least squares (see **Chapter 4** for details), Algebraic Sub-grid Scales, Finite Calculus or Orthogonal Sub-scales) lead to an introduction of a pressure Laplacian scaled with an algorithmic stabilization parameter τ in the lower diagonal position. It is interesting to mention that the first order fractional step method [29] also introduces some pressure stability as the split by itself brings in the Laplacian operator applied to the pressure p_n and scaled with the time step Δt, which assumes from this point of view the role of the stabilization parameter. Intuitively it is clear that the second order fractional step method looses the stability properties as there the Laplacian becomes applied to the pressure difference $p_{n+1} - p_n$ instead.

Quasi-incompressible case In the quasi-incompressible formulation no problems associated to the pressure stability are encountered as long as one uses low to moderate values of bulk modulus. However when using large values of bulk modulus and thus approaching the incompressible regime, the instable pressure behavior becomes an issue. One can see that for a high values of the bulk modulus the tangent matrix in the Eq. (3.29) obtains the following form

$$\begin{pmatrix} \rho\frac{\mathbf{M}}{\Delta t^2} - \mu\frac{\mathbf{L}}{\Delta t} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{pmatrix} \tag{3.35}$$

which is identical to the one for a purely incompressible case.

For pressure-velocity pairs that do not pass the inf-sup condition [25], such matrix is known to become more and more ill-conditioned (eventually indefinite for purely incompressible problems).

This in turn implies, by definition of ill-conditioned matrix, that small perturbations in the residuals will lead to arbitrarily large variations in the solution, thus implying that small perturbations in the residuals may eventually grow and destroy the pressure solution.

In order to introduce and provide an heuristic justification of our proposed modification, let us consider a static equilibrium condition under the assumption of zero body force $\mathbf{F} = \mathbf{0}$, and let's assume further that the nodes are distributed in a Cartesian way.

Static solution implies that the variation of displacement from one time step to the following is zero. On the other hand at equilibrium $p_n \equiv \bar{p}_{n+1}^{i+1}$ and $\bar{V}_{n+1}^{i+1} \equiv \bar{V}_n$. We can now make the following observations:

- Since the operator $\mathbf{M}_p$ acts on both $\bar{p}_{n+1}$ and $\bar{p}_n$, *independently of the spatial distribution of pressure*, a pressure that does not vary in time will guarantee that the pressure residual will be identically equal to zero (of course in the hypothesis that the volume does not change).

- For the Cartesian distribution assumed, different spatial pressure distributions exist that make zero the residual of the momentum equation, namely a constant (zero) pressure and a checkboard mode [53].

These last two observations are enough to prove that multiple solutions may exist, unless $\mathbf{M}_p$ is modified so to introduce some spatial relation between the pressure on neighboring nodes. For incompressible fluids the pressure is commonly stabilized by the introduction of a "pressure Laplacian". Even if this is optimal in the context of monolithic techniques, we do not follow this approach in current work, since we like the pressure term to be diagonal.

Our proposal to palliate the problem is to modify Eq. (3.22) as

$$\bar{\mathbf{r}}_p = \mathbf{M}_p^c \bar{p}_n - \mathbf{M}_p \bar{p}_{n+1}^i - \kappa \left(\bar{V}_{n+1}^{i+1} - \bar{V}_n \right) \tag{3.36}$$

where $\bar{p}_n$ is weighted by a consistent pressure mass matrix rather that by the lumped version. The key point here is that, at steady state Eq. (3.36) simplifies to

$$\left(\mathbf{M}_p - \mathbf{M}_p^c \right) \bar{p} \tag{3.37}$$

where $\bar{p} = \bar{p}_{n+1}^{i+1} = \bar{p}_n$ is the steady state pressure. The introduced modification does not introduce any inconsistency as for $h \to 0$, the consistent and lumped mass matrices coincide.

The important issue from the point of view of the pressure stability is that the modified form

introduces an additional requirement on the spatial distribution of the pressure, that is now called to guarantee that $(\mathbf{M}_p - \mathbf{M}_p^c)\bar{p} = 0$.

Since the operator $\mathbf{M}_p^c - \mathbf{M}_p$ approximate the Laplacian [7] (in 1D it can be shown to be exactly equivalent), the proposed stabilization can be thus viewed (at steady state) as a simple Laplacian-type stabilization, ensuring some degree of pressure stability .

Remark: The stabilization effect achieved is still weak for very high values of the bulk modulus. This can be easily seen by considering that the "stabilization term" is weighted by $\frac{1}{\kappa}$ which implies that $\kappa \to \infty$ the instable equation is recovered.

We argue nevertheless that the modification proposed in very effective in the practice, for a large range of the bulk modulus values.

3.3.4 Global Pressure condensation

Up to this moment we have to defined a monolithic system of equations which "tightly" relates the displacements of the nodes with the pressure. By "tight" relation we mean that the system is fully coupled, namely none of the entries of the tangent stiffness is zero, thus requiring simultaneous solution for the displacements and the pressure. Such system can be easily constructed and allows in principle the use of any stabilization technique. However the monolithic system defined by Eq. (3.29) does not provide any significant advantage in comparison with the standard incompressible one. The poor conditioning even for the single field problems would require utilization of direct solvers.

In order to deal with large problems we take advantage of the introduced compressibility and symbolically condense the pressure from the global system in order to obtain a displacement-only system that is equivalent to the original linearized system. As we shall see later, this is possible if pressure is stabilized using the method described above, while it would be impossible if a Laplacian type stabilization was used.

The symbolic condensation of the system in Eq. (3.29) gives

$$\left(\rho\frac{\mathbf{M}}{\Delta t^2} - \mu\frac{\mathbf{L}}{\Delta t} + \kappa\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D}\right) d\bar{\mathbf{u}} = \bar{\mathbf{r}}_m\left(\bar{\mathbf{u}}_{n+1}^i, \bar{p}_{n+1}^i\right) + \mathbf{G}\mathbf{M}_p^{-1}\bar{r}_p\left(\bar{\mathbf{u}}_{n+1}^i, \bar{p}_{n+1}^i\right) \tag{3.38}$$

This is practically implemented as the equivalent multi-step procedure:

1. Given the pressure $\bar{p}_n$ and $\Delta\bar{\mathbf{u}}_{n+1}^i$ (and consequently the nodal volume $\bar{V}^i$), compute the

pressure $\bar{p}_{n+1}^{i+1}$ such that

$$\bar{r}_p\left(\bar{\mathbf{u}}_{n+1}^{i}, \bar{p}_{n+1}^{i+1}\right) = \mathbf{0}$$

(note that the position of the nodes is kept fixed)

2. Given $\bar{p}_{n+1}^{i+1}$ as obtained in the first step, compute $\bar{\mathbf{r}}_m\left(\bar{\mathbf{u}}_{n+1}^{i}, \bar{p}_{n+1}^{i+1}\right)$ and solve the system

$$\left(\rho\frac{\mathbf{M}}{\Delta t^2} - \mu\frac{\mathbf{L}}{\Delta t} + \kappa\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D}\right) d\bar{\mathbf{u}} = \bar{\mathbf{r}}_m\left(\Delta\bar{\mathbf{u}}^{i}, \bar{p}_{n+1}^{i}\right)$$

3. Update the nodal position as

$$\bar{\mathbf{x}}_{n+1}^{i+1} = \bar{\mathbf{x}}_{n+1}^{i} + d\bar{\mathbf{u}}$$

4. Check convergence in terms of $\bar{\mathbf{u}}$. If not converged go back to step 1.

The only system of equation to be solved is thus found in the second step. Introducing the notation $\mathbf{H}_f = \rho\frac{\mathbf{M}}{\Delta t^2} - \mu\frac{\mathbf{L}}{\Delta t} + \mathbf{G}\mathbf{M}_p^{-1}\mathbf{D}$ to denote the tangent stiffness of the fluid corresponding to the momentum equation we can write

$$\mathbf{H}_f d\bar{\mathbf{u}} = \bar{\mathbf{r}}_m \tag{3.39}$$

which is written exclusively in terms of displacements.

Remark Note that pressure condensation did not require solution of any system of equations due to the lumped format of the pressure mass matrix. In the case a classical pressure stabilization matrix was added to the monolithic system, such global condensation would have been impossible due to the enormous effort of symbolically inverting a non-diagonal sparse matrix (in the case of discrete Laplacian $\mathbf{D}\mathbf{M}_v^{-1}\mathbf{G}$) or impossibility of inversion of the continuous Laplacian $\mathbf{L}$.

Before proceeding to the section describing the coupling with the structure and presenting the entire monolithic FSI solution algorithm, we would like to outline the similarities and emphasize the differences between the approach proposed here and the other quasi-incompressible or penalty formulations.

3.4 Quasi-incompressibility and penalty methods

The proposed methodology can be viewed as a type of penalty method. Even though the governing equations system defined by 3.15 and 3.22 were derived using a physical argument here, it is possible to obtain them from an abstract starting point, namely viewing the bulk modulus as a parameter used

for penalizing the continuity equation. This corresponds to reformulating the momentum equation constrained by the incompressibility condition into an unconstrained problem, see e.g. [54], [58]. In the next section we present a brief comparison of the proposed methodology with the classical penalty method used for imposing incompressibility in fluid and solid mechanics. This will enable us to highlight the similarities as well as the differences.

Penalty method The classical element-based penalty method defines what we have called "pressure constitutive relation" that is equivalent at the continuous level to Eq. (3.8). However on the discrete level the pressure is condensed element-wise. Elemental pressure condensation for linear triangles/tetrahedra results in the discontinuous (constant element-wise) pressure field. The residual $\mathbf{r}_{penalty}$ and tangent stiffnesses $\mathbf{H}_p$ for the penalty method in conjunction with Backward Euler time integration give

$$\mathbf{H}_{penalty} = \mathbf{M}\frac{1}{\Delta t^2} + \nu\mathbf{L}\frac{1}{\Delta t} + \mathbf{A} \tag{3.40}$$

$$\bar{\mathbf{r}}_{penalty} = \mathbf{F}_{n+1} + \mathbf{F}^p_{n+1} - \mathbf{M}\frac{\bar{\mathbf{u}}^i_{n+1} - \bar{\mathbf{u}}_n}{\Delta t^2} - \mu\mathbf{L}\frac{\bar{\mathbf{u}}^i_{n+1} - \bar{\mathbf{u}}_n}{\Delta t} \tag{3.41}$$

with $\mathbf{A} = \int_\Omega \mathbf{B}^T\mathbf{C}k\mathbf{B}\,d\Omega$ and $\mathbf{F}^p = \int_\Omega \mathbf{B}^T\bar{p}$, where $\mathbf{B}$ is the standard strain-displacement matrix (see e.g. [7]) and $\mathbf{C}_K$ is the "volumetric" constitutive tensor.

$$\mathbf{C}_K = \frac{1}{3}\begin{pmatrix} \kappa & \kappa & \kappa & 0 & 0 & 0 \\ \kappa & \kappa & \kappa & 0 & 0 & 0 \\ \kappa & \kappa & \kappa & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3.42}$$

The linearized system then reads

$$\mathbf{H}_p d\bar{\mathbf{u}} - \bar{\mathbf{r}}_{penalty} = 0 \tag{3.43}$$

Eq. (3.43) is solved for $\bar{\mathbf{u}}^{i+1}_{n+1} = \bar{\mathbf{u}}^i_{n+1} + d\bar{\mathbf{u}}$ and pressure is recovered element-wise as soon final $\bar{\mathbf{u}}_{n+1}$ is obtained (note the difference with the method proposed here, where pressure is updated at every non-linear iteration).

Classical penalty methods with elementally condensed pressure are computationally cheap. They were successfully applied to the fluid-structure interaction in [21]. Its application to fluid-membrane interaction was presented in [59].

Unfortunately classical penalty methods based on discontinuous pressure lead to volumetric locking for simplicial elements when increasing the bulk modulus. A good illustration and explanation of locking can be found in [53]. In practice already for $\kappa \approx 100000$ [Pa] locking occurs for coarse meshes. This corresponds to the speed of the sound wave of ≈ 10 [m/s], which gives an approximate bound over the applicability of the standard penalty method. The goal pursued here was to maintain the advantages of the penalty methods for FS coupling, but to increase the upper bound for the value of bulk modulus in order to simulate a wider range of problems. The use of continuous pressure interpolation in conjunction with the computation of the pressure according to Eq. (3.20) proposed here reduces the number of constrains imposed by associating the pressure with the nodal volume/area change, in contrast to elemental volume/area change, and does not lead to locking. The minimization of the computational cost associated with the pressure condensation on the global level arising is addressed in 3.5.1.

Finally we would like to mention the iterative penalty method proposed by Codina [60] for stationary Stokes and Navier-Stokes equations. The basic idea of this method is solving the penalized equations at each iteration but adding the right-hand side term that is basically the residual of the continuity equation of the previous iterate. On the continuous level the pressure is computed as

$$p_{n+1}^{i+1} = p_{n+1}^{i} - \kappa \nabla \cdot \mathbf{v}_{n+1}^{i} \tag{3.44}$$

where $\mathbf{v}_{n+1}^{i}$ is the velocity at the previous non-linear iteration. The final goal of that approach is to converge to the truly incompressible solution.

This is essentially differs from the approach followed here, as permitting the slight compressibility is necessary for the fluid-structure coupling purposes. In fact in the method proposed here our goal is the utilization of the smallest compressibility constant, that is sufficiently large to represent the desired physical phenomenon. As we shall see in the examples section, extremely large values of the bulk modulus lead to a deteriorated computational efficiency of the method. Additionally, the pressure in the momentum equation is not condensed in [60], as the FSI applications are not considered there, defining another important difference with our methodology.

3.5 Coupling with the structure

In the following "monolithic FSI" means a discrete equation system that includes the equations of both the sub-domains. It should not be confused with the term "monolithic single-field" that

was used when describing the fluid formulation, where it referred to simultaneous solution for the primary variables.

Here we shall present the coupling strategy with the structure and establish the monolithic FSI system. The illustration will be performed without specifying any particular type of structure. It can be a solid, membrane, a shell or any other FE structure. The only prerequisite is that the discrete equations are written in terms of displacements. We note, that the time integration scheme must be consistent with the one used for the fluid. The discrete momentum equations for the solid in the absence of damping, using Backward Euler time integration can be written as

$$\left(\rho\frac{\mathbf{M}}{\Delta t^2}+\mathbf{K}\right)d\bar{\mathbf{u}}=\bar{\mathbf{r}}_s \tag{3.45}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ is the stiffness matrix, $\mathbf{F}$ is the external force vector, $\bar{\mathbf{a}}_{n+1}=\frac{\bar{\mathbf{u}}_{n+1}-2\bar{\mathbf{u}}_n-\bar{\mathbf{u}}_{n-1}}{\Delta t^2}$ the acceleration and $\bar{\mathbf{u}}$ is the total displacement. For applying Newton's method we define the dynamic residual and tangent stiffness

$$\bar{\mathbf{r}}_s=\mathbf{F}-\mathbf{M}\frac{\bar{\mathbf{u}}^i_{n+1}-\bar{\mathbf{u}}_n}{\Delta t^2}-\mathbf{K}\left(\bar{\mathbf{u}}^i_{n+1}\right) \tag{3.46}$$

$$\mathbf{H}_s=\frac{\partial\bar{\mathbf{r}}_s}{\partial\bar{\mathbf{u}}}=\rho\frac{\mathbf{M}}{\Delta t^2}+\mathbf{K} \tag{3.47}$$

where subscript s stands for "structure", superscript "i" stands for Newtons' iteration index.

The tangent matrix and the residual for the fluid given by Eq. (3.39) and Eq. (3.15) will be distinguished by subscript "f". We shall split the dynamic tangent stiffness of the fluid into two parts for reasons that will become clear later.

$$\mathbf{H}_f=\mathbf{H}^a_f+\mathbf{H}^b_f \tag{3.48}$$

where

$$\mathbf{H}^a_f=\mathbf{M}-\mu\mathbf{L} \tag{3.49}$$

and

$$\mathbf{H}^b_f=-\kappa\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D} \tag{3.50}$$

The linearized monolithic FSI equations system is obtained then in two steps: first a standard FE assembly procedure is carried out by looping over all the elements (fluid and structure). Structure elements contribute $\bar{\mathbf{r}}_s$ and $\mathbf{H}_s$ whereas fluid elements contribute $\bar{\mathbf{r}}_f$ and $\mathbf{H}^a_f$ to the unique FSI

dynamic residual and tangent stiffness $\bar{\mathbf{r}}_{FSI}$ and $\mathbf{H}_{FSI}$, respectively.

Then $\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D}$ needs to be added. The monolithic FSI linearized system can then be written as

$$\left(\mathbf{H}_{FSI} - \kappa\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D}\right) d\bar{\mathbf{u}} = \bar{\mathbf{r}}_{FSI} \tag{3.51}$$

Note that the $\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D}$ is a global matrix defined for the fluid only. Its entries corresponding to the degrees of freedom of the structure nodes are zero.

Remark Note that in the proposed monolithic approach the interface forces, i.e. the Neumann part of the interface boundary conditions (fluid pressure acting on the solid) are automatically accounted for as the pressure of the interface node (that is shared by the fluid and the structure) is added to the residual of the momentum equation at the next iteration. [21].

3.5.1 Computational efficiency

The addition of the $\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D}$ term is a computationally expensive procedure. The costs associated with construction and storage of this matrix at each Newton's iteration would make the method unfeasible for practical applications. This is easy to understand as the resulting matrix is densely populated and its computation involves sparse matrix-matrix products. We propose to treat the problem defined by Eq. (3.51) by an iterative technique. The choice of using Krylov-type methods implies that a matrix-free approach can be used, thus removing the need of constructing and storing of the computationally expensive $\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D}$ term.

We shall illustrate the methodology using diagonally preconditioned Conjugate Gradient method (CG), which is actually used in the implementation of this work.

The main idea of matrix-free approaches is that only the product $\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D}d\bar{\mathbf{u}}$ needs to be known, and this can be done without computing matrix-matrix products and and storing the matrix $\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D}$.

This can be easily obtained by a sequence of matrix-vector products in the form, avoiding evaluation and storage of any matrix-matrix product:

1. $\mathbf{e}_1 = \mathbf{D}d\bar{\mathbf{u}}$
2. $\mathbf{e}_2 = \mathbf{G}\mathbf{e}_1 = \mathbf{D}^T\mathbf{e}_1$
3. $\mathbf{e}_3 = \mathbf{M}_p^{-1}\mathbf{e}_2$

Table 3.1: Number of non-zero entries and respective matrix sizes

parameter	$\mathbf{GM}_p^{-1}\mathbf{D}$	$\mathbf{D}$
$n_{non-zero}$ per node in 2D	18	6
$n_{non-zero}$ per node in 3D	124	26
size in 2D	$2N \cdot 2N$	$N \cdot 2N$
size in 3D	$3N \cdot 3N$	$N \cdot 3N$
$n_{non-zero}$ total in 2D	$18{\cdot}2N \cdot 2N$	$6{\cdot}2N \cdot N$
$n_{non-zero}$ total in 3D	$124{\cdot}3N \cdot 3N$	$6{\cdot}3N \cdot N$

Taking advantage that $\mathbf{G} = \mathbf{D}^T$ we thus need only to construct and store the discrete divergence operator $\mathbf{D}$. Ultimately instead of computing three matrix-matrix products we end up computing two matrix vector products (assuming $\mathbf{M}_p$ being a lumped matrix). Apart from the memory efficiency of this approach, it also turns out that, due to the structure of the $\mathbf{D}$ operator, the application of this technique requires less operations than the application of the assembled matrix. The speed up obtained by doing this in comparison with direct computation of $\mathbf{GM}_p^{-1}\mathbf{D}$ results to be of ~ 3 in 2D and ~ 7 in 3D. This speed up refers to the solution of linearized system performed at each non-linear iteration. Details of the speed up calculation are presented next:

Speed-up estimation Here the estimation is presented whose results were mentioned in Section 3.5.1. We shall compare the number of floating point operations, resulting when constructing the $\mathbf{GM}_p^{-1}\mathbf{D}$ matrix and when using the matrix-free method that involves only subsequent matrix-vector multiplications. Note that in both cases $\mathbf{M}_p$ is taken in the lumped format.

Table 3.1 summarizes the data necessary for the FLOPs estimation in both cases under the assumption of structured mesh. The number of non-zero entries $n_{non-zero}$ is estimated taking into account the structure of $\mathbf{GM}_p^{-1}\mathbf{D}$ and $\mathbf{D}$ matrices. In the former one each node contributes to the number of non-zero entries equal to the number of the neighbors of the neighbors of the node ("second neighbors") multiplied with the matrix size (3N*3N in 3D and 2N*2N in 2D). In the latter one, it is equal to the number of the nodal neighbors only, multiplied by the matrix size (3N*N in 3D and 2N*N in 2D). In the estimation of the total number of the non-zero entries, we do not consider the effect of the boundary nodes, which will slightly decrease the total number of non-zero entries. The total number of non-zero entries is therefore the product between the non zero contributions of a node and the matrix size, N is the number of nodes in the discretization. Finally, Table 3.2 presents the cumulative FLOPS for either case. Note, that in the matrix free method we need to perform 3 steps of 3.5.1, where step 1 and step 3 lead to an equal number of FLOPS since $\mathbf{G} = \mathbf{D}^T$ and step 2 leads to N FLOPS, as $\mathbf{M}_p^{-1}$ is a lumped matrix of size N.

The matrix-free procedure leads to a speed-up of ≈ 3 in 2D and of ≈ 7.2 in 3D in comparison

Table 3.2: Comparison of the number of the floating point operations

parameter	$\mathbf{GM}_p\mathbf{D}$	Matrix free	speed-up
nFLOPs in 2D	$72{\cdot}N^2$	$24{\cdot}N^2+N$	3
nFLOPs in 3D	$1116{\cdot}N^2$	$156{\cdot}N^2+N$	7.2

with the direct construction of the $\mathbf{GM}_p^{-1}\mathbf{D}$ matrix.

Coupled problem solution algorithm In Eq. (3.29) the displacements of the interface nodes (i.e. the ones that belong both to the fluid and the solid) are automatically multiplied by the sum of the respective contributions of the fluid and solid dynamic tangent stiffnesses. The system is then treated exactly in the same way as a single field one, i.e. the solver does not "see" the difference between the fluid and the solid and no interface equation needs to be specified. The entire solution procedure is summarized in Table 3.3

We see, that the two conditions that need to be satisfied in the FSI solution, namely

- equilibrium of forces along the interface boundary
- continuity of displacements along the interface boundary

are satisfied automatically. The computational cost introduced by iterations for satisfaction of these two conditions in a partitioned approach is shifted to the quality of convergence in a monolithic approach.

Table 3.3: Implementation procedure for the solution of the monolithic FSI involving quasi-incompressible updated Lagrangian fluid (ULF)

1. Start the non-linear loop
   - update the pressures of the fluid elements using Eq. (3.20)
   - assemble the dynamic residual $\bar{\mathbf{r}}_{FSI}$ in standard FE manner
     if solid element use Eq. (3.46)
     if fluid element use Eq. (3.15)
   - assemble the dynamic tangent stiffness $\mathbf{H}_{FSI}$ in standard FE manner
     if solid element use Eq. (3.47)
     if fluid element use Eq. (3.49)
   - solve linear system $\left(\mathbf{H}_{FSI}+\kappa\mathbf{GM}_p^{-1}\mathbf{D}\right)d\bar{\mathbf{u}}=\bar{\mathbf{r}}_{FSI}$
   - update displacements $\bar{\mathbf{u}}_{n+1}^{i+1}=\bar{\mathbf{u}}_{n+1}^{i}+d\bar{\mathbf{u}}$
2. repeat until convergence in displacements is achieved
3. re-mesh

Remark The computational cost associated with the solution of each time step will depend on two factors:

- Number of non-linear iterations
- Time spent per one non-linear iteration or, equivalently number of the linear solver iterations

These two criteria will be considered in the assessment of the method presented in the next section of the Chapter.

Remark 2 Time integration and linearization were presented in conjunction with the Backward Euler algorithm for the sake of clarity. However in the implementation of the present work a Newmark-Bossak scheme was used. The Newmark algorithm defines a second-order accurate unconditionally stable implicit time integration scheme. Bossak modification provides the improvement for highly geometrically non-linear problems, where classical Newmark method exhibits lack of stability. Bossak scheme introduces numerical dissipation, thus damping the high frequencies. We propose to use a Newmark-Bossak scheme for the time integration of fluid equations aiming at the monolithic coupling with structures. Newmark-Bossak scheme is widely used in the structural codes and we "adapt" the fluid formulation to fit the former one.

The Newmark-Bossak formulae for the acceleration and velocity read

$$\bar{\mathbf{v}}_{n+1} = \frac{\bar{\mathbf{u}}_{n+1} - \bar{\mathbf{u}}_n}{\Delta t} - \left(\frac{\gamma}{\beta} - 1\right)\bar{\mathbf{v}}_n - \frac{\Delta t}{2}\left(\frac{\gamma}{\beta} - 2\right)\bar{\mathbf{a}}_n \tag{3.52}$$

$$\bar{\mathbf{a}}_{n+1} = \frac{\bar{\mathbf{u}}_{n+1} - \bar{\mathbf{u}}_n}{\beta\Delta t^2} - \frac{1}{\beta\Delta t}\bar{\mathbf{v}}_n - (\frac{1-2\beta}{2\beta})\bar{\mathbf{a}}_n \tag{3.53}$$

where $\gamma = \frac{1}{2} - \alpha^B$ and $\beta = \frac{(1-\alpha^B)^2}{4}$ with $\alpha^B > 0$. If $\alpha^B = 0$ we obtain the standard Newmarks scheme at its optimal, without any numerical dissipation. Numerical dissipation control is obtained by varying the coefficient α^B [53]. Substituting Eq. (3.52) into Eq. (3.15) and subsequent linearization leads to the following tangent stiffness $\mathbf{H}_B$ and residual $\bar{\mathbf{r}}_B$.

$$\mathbf{H}_B = \rho(1-\alpha^B)\mathbf{M}\frac{1}{\beta\Delta t^2} - \mu\mathbf{L}\frac{\gamma}{\beta\Delta t} - \rho c^2\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D} \tag{3.54}$$

$$\bar{\mathbf{r}}_B = \mathbf{F}_{n+1} + \mathbf{G}\bar{p}_{n+1} - \rho\mathbf{M}\left(\frac{(1-\alpha^B)\bar{\mathbf{u}}_{n+1} + \alpha^B\bar{\mathbf{u}}_n}{\beta\Delta t^2} - \frac{\bar{\mathbf{v}}_n}{\beta\Delta t} - (\frac{1-2\beta}{2\beta})\bar{\mathbf{a}}_n\right) + \mu\mathbf{L}\left(\frac{\bar{\mathbf{u}}_{n+1} - \bar{\mathbf{u}}_n}{\Delta t} - \left(\frac{\gamma}{\beta} - 1\right)\bar{\mathbf{v}}_n\right) \tag{3.55}$$

3.6 Examples

This section shows functionality of the formulation.

3.6.1 Dam break example

Practical applicability

The quality of a monolithic method is strongly related to the convergence characteristics it provides. The proposed formulation does not suffer from volumetric locking in comparison with the quasi-incompressible formulations or penalty methods that use discontinuous pressure. Therefore the bounds that can provide us the practically admissible values of bulk modulus are related to the convergence speed that will be discussed here. Two types of convergence are considered. First is the convergence of Newton method, which reflects the quality of the proposed linearization of the coupled system. Second issue is the convergence of the iterative solver.

The latter is expected to be affected by the value of the bulk modulus to a great extent. Generally when penalty methods are used, the conditioning of the system deteriorates with increasing of the penalty constant. It is important to estimate how much the value of the bulk modulus affects the convergence speed of the CG linear solver. The convergence results presented next do not pretend to be exhaustive or to provide a clear limit for the value of the bulk modulus one can use. Rather they show the tendency and give a qualitative outlook of the convergence behavior.

We shall provide the convergence results obtained when solving the dam break problem and water column falling against an elastic obstacle as proposed by Hübner et al. [52]. These problems are depicted on Fig. 3.1. The properties of the elastic obstacle are as follows:

- width w = 0.012 [m]
- height h = 0.08 [m]
- density $\rho = 2500$ [kg/m^3]
- Young's modulus E= 10^6 [Pa]
- Poisson's ratio $\nu = 0$

Dimensions of the water column are: w=0.146 [m] and height H=0.292 [m]. The distance between the walls is b=0.584 [m].

This example serves us for judging to which extent the coupling with the structure slows down the computation in comparison with the case containing fluid only. Four different values of bulk

(a) Fluid only (b) FSI

Figure 3.1: Dam break problems

κ	c	N-R (fluid)	CG (fluid)	N-R (FSI)	CG (FSI)
10E05	10	3.96	25	4.12	63
10E06	33	4.24	68	4.2	115
10E07	100	4.59	200	4.89	323
10E08	333	4.91	460	4.93	816

Table 3.4: Convergence characteristics: Δt=0.001 [s], 1000 elements

κ [Pa]	c [m/s]	N-R (fluid)	CG (fluid)	N-R (FSI)	CG (FSI)
10E05	10	4.1	160	4.3	550
10E06	33	4.42	345	4.51	1027
10E07	100	4.7	1010	4.92	2015
10E08	333	4.92	2220	4.95	5200

Table 3.5: Convergence characteristics: Δt=0.001 [s], 30000 elements

modulus were used. Two meshes were considered, one containing $\approx$ 1000 linear triangles and the other $\approx$ 30000 linear triangles. The convergence criteria were set to $\frac{||d\bar{\mathbf{u}}||}{||\bar{\mathbf{u}}_{n+1}||}$ <10E-06 for the non-linear loop and $\frac{||\bar{\mathbf{r}}-\mathbf{H}d\bar{\mathbf{u}}_{\Phi}||}{||\bar{\mathbf{r}}-\mathbf{H}d\bar{\mathbf{u}}_{0}||}$ <10E-08 for the Conjugate Gradient linear solver. Iterations number (averaged over the simulation time span 0-1 [s]) for both the non-linear loop and linear solver are given for both the purely fluid problem and the FSI problem in the Table 3.4 (rough mesh) and Table 3.5 (fine mesh). The notation is as follows: c is the sound speed corresponding to the bulk modulus κ, "N-R" and "CG" are the number of non-linear (Newton-Raphson) and linear solver (CG) iterations respectively. "Fluid" refers to the purely fluid problem, "FSI" refers to water column against elastic obstacle problem.

These results prove that the convergence of the non-linear (Newton-Raphson) loop does not deteriorate much as the value of the bulk modulus increases. Basically two "ingredients" of the dynamic tangent are competing - the inertia term (inversely proportional to the time squared) and the volumetric term. We assume that the influence of the viscous term upon the non-linear convergence behavior is negligible for low values of viscosity such as that of water. Small time steps lead to domination of the inertia term $\frac{1}{\Delta t^2}\mathbf{M}$ in the tangent stiffness, whereas high values of the bulk modulus increase the impact of the volumetric term $\kappa\mathbf{G}\mathbf{M}_p^{-1}\mathbf{D}$. As mentioned in 3.3.2, the inertia term is linearized exactly, as the mass matrix does not change within one time step.

Note that the linearization that lead to the definition of the monolithic system in Eq. (3.29) is not entirely exact as the exact linearization would imply (for example) the differentiation of the $\mathbf{G}$ operator with respect to the displacement of the nodes.

Nevertheless the optimal convergence rates of Newton-Raphson method are assured if the time step is sufficiently small.

Results reported in Tables 3.4 and 3.5, confirm that the optimal convergence speed for the non-linear problem is retained both for the case of fluid only and considering the complete FSI problems. This indicates that the linearization of the monolithic system is done correctly, and in particular the presence of the structure and thus the interaction does not deteriorate the convergence of the non-linear procedure. For both coarse and fine discretizations convergence is achieved within $\sim$3-4 non-linear iterations in average.

Results also show how the number of iterations of the linear solver grows as κ increases, which is associated with the deteriorating conditioning of the linear system to be solved. The results provided here indicate that a preconditioner better than the one utilized here (diagonal preconditioner) could improve the convergence rates of the linear solver.

It is also important to observe that the number of iterations needed for the convergence of the

CG differs in the FSI problem (where an elastic obstacle is included) by a factor of $\sim 2 - 2.5$ from that of the fluid-only problem for both meshes.

Remark: As mentioned above it is possible to set-up and solve directly a monolithic fluid displacement-pressure system according to Eq. (3.29). The implementation of this formulation has shown that the Conjugate Gradient solver with diagonal preconditioner fails to resolve the problem already for a bulk modulus of $\propto$100000 [Pa] with the mesh and time step size equivalent to the one used above. Only the application of a direct solver (Super LU was used for this verification in this work) circumvents the problem for practically any value of the bulk modulus. Since the direct solvers generally cannot be applied to real-life problems involving hundreds of thousands or more degrees of freedom, this fact shows the advantage achieved when pressure condensation is applied. The implementation of the monolithic system also allowed verifying that the result obtained by the monolithic approach are identical to the ones achieved by our pressure condensation algorithm.

Comparison with other methods

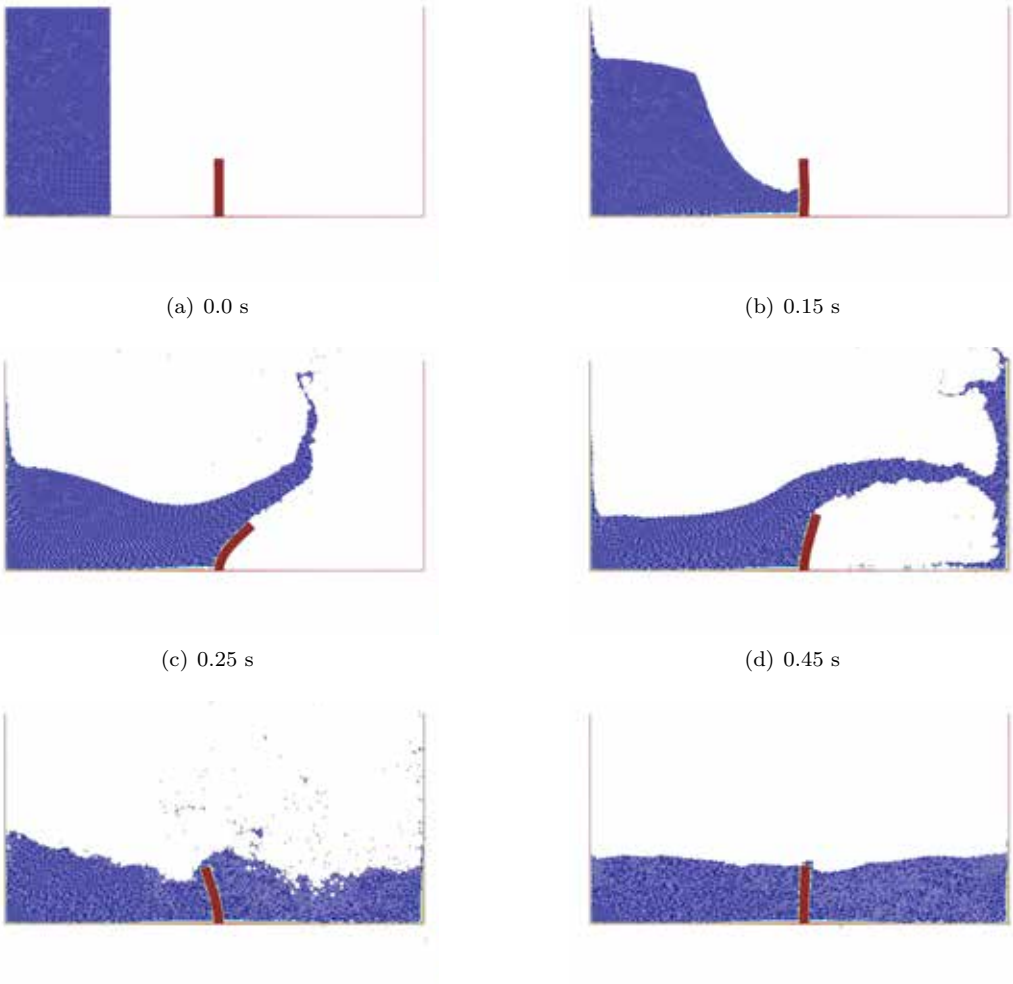
Here we shall compare the results obtained using the present formulation with the published results obtained by different methods. The comparison is made considering the temporal evolution of the horizontal displacement d_x of the upper left corner of the elastic obstacle.

Fig. 3.2 presents snapshots of the simulation using the proposed method, taken at various time instances.

Solutions of this problem obtained in [21] and with the approach proposed here are presented in Fig. 3.3. Bulk modulus for modeling quasi-incompressibility behavior was set to $\kappa = 10^5$ [Pa] for the unified formulation [21] and $\kappa = 10E07$ [Pa] for the formulation proposed here. Fig. 3.3 depicts the temporal evolution of the horizontal displacement component of the upper right corner of the obstacle. The total time in the simulation was 2.5 [s]. Good agreement is obtained with results given in [21].

Another comparison is made with the Eulerian monolithic approach with level-set method for the free surface identification published in [41]. There water is modeled as incompressible and the obstacle as a hyper-elastic material. The comparison is depicted in Fig. 3.4.

Finally, the effect of the bulk modulus upon the solution of the problem has been studied. In Fig. 3.5 the displacement evolution of the upper right corner of the elastic obstacle obtained using different values of the bulk modulus is shown. One can see good agreement between the graphs. In particular, this ensures that the method is fairly insensitive to the value of the bulk modulus in the sense of representing the quasi-incompressibility. This means that as long as the bulk modulus is high enough



(a) 0.0 s

(b) 0.15 s

(c) 0.25 s

(d) 0.45 s

(e) 1.0 s

(f) 2.5 s

Figure 3.2: Dam break against an elastic obstacle

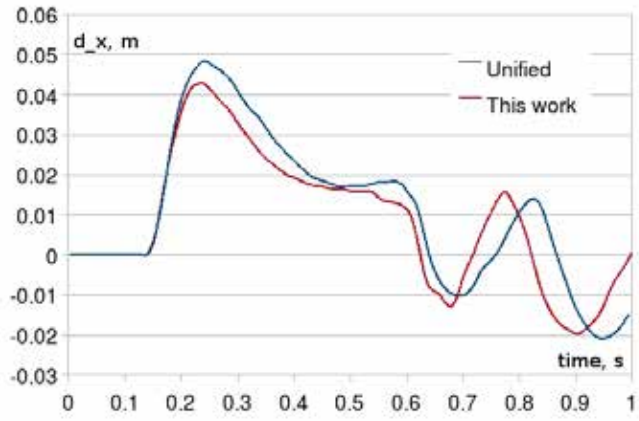


Figure 3.3: Dam break against an elastic obstacle. Comparison of the current approach (continuous line) with the results of the unified formulation (dashed line)

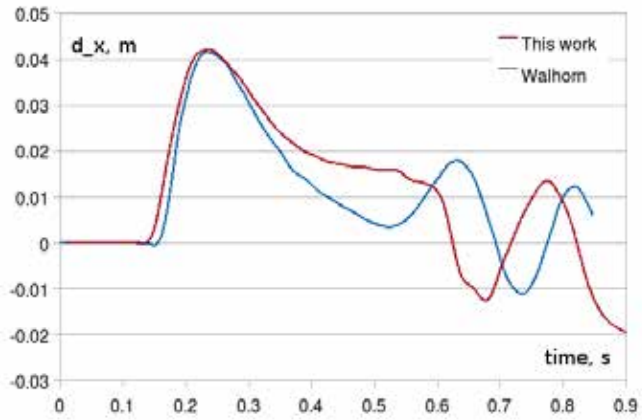


Figure 3.4: Dam break against an elastic obstacle. Comparison of the proposed approach (continuous line) with the Eulerian monolithic approach (dashed line)

to prevent a large volume variation and ensures that the associated finite sound wave propagation speed is sensibly higher than the characteristic velocity of the problem, the approximation made can be accepted. Note that to obtain maximal performance, one must consider on one side that the bulk modulus must be high enough for the correct representation of the physics, but on the other hand it should be kept as small as possible to ensure the fast convergence of the iterative solvers.

Pressure stability Finally we present the results reflecting the pressure stability. In Fig. 3.6 we show the pressure distribution patterns at different time instances (0.05, 0.2 and 0.3 s]) for the

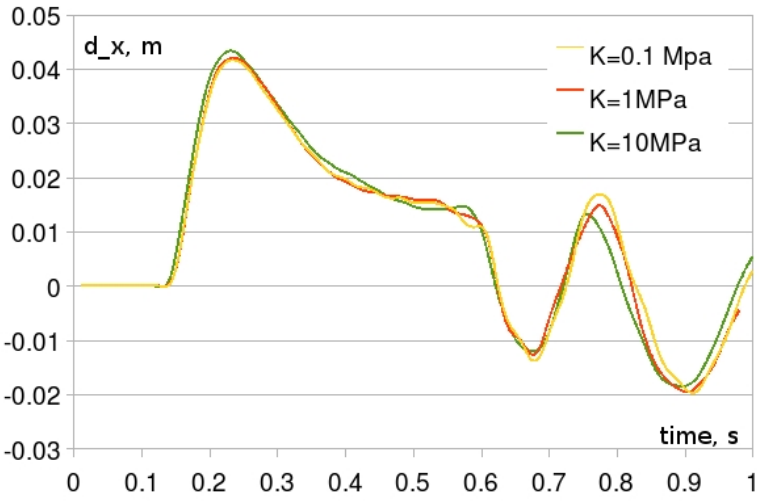


Figure 3.5: Dam break against an elastic obstacle - Comparison of the results obtained for different values of bulk modulus

dam break example for $\kappa = 0.5$ [MPa]. The results obtained when the pressure is stabilized by the difference between the consistent and lumped mass matrices is compared with the solution where both matrices are used in the lumped format.

One can clearly see the spurious (checkerboard-like) modes exhibited already at $t = 0.05$ [s] in case no stabilization is applied. We can notice that already for a medium value of the bulk modulus pressure stability of the quasi-incompressible formulation becomes an important issue. The results corresponding to the stabilization proposed in the current work provide an empirical proof of the functionality of the proposed stabilization.

3.6.2 Deformation of an elastic plate subjected to water pressure

Next example presents the deformation of an elastic plate subjected to water pressure. A water container of width $A = 0.1[m]$ with water level $L = 0.14[m]$ is closed by an elastic cover of height $H = 0.079[m]$ and width $s = 0.005[m]$, which is fixed at the top (Fig. 3.7) . The cover is released and exposed to the water pressure, which induces deformation. The elastic cover is modeled with the following properties

- density $\rho = 1100$ $[kg/m^3]$
- Young modulus $E = 0.014$ [GPa]
- Poisson ratio $\nu = 0.4$

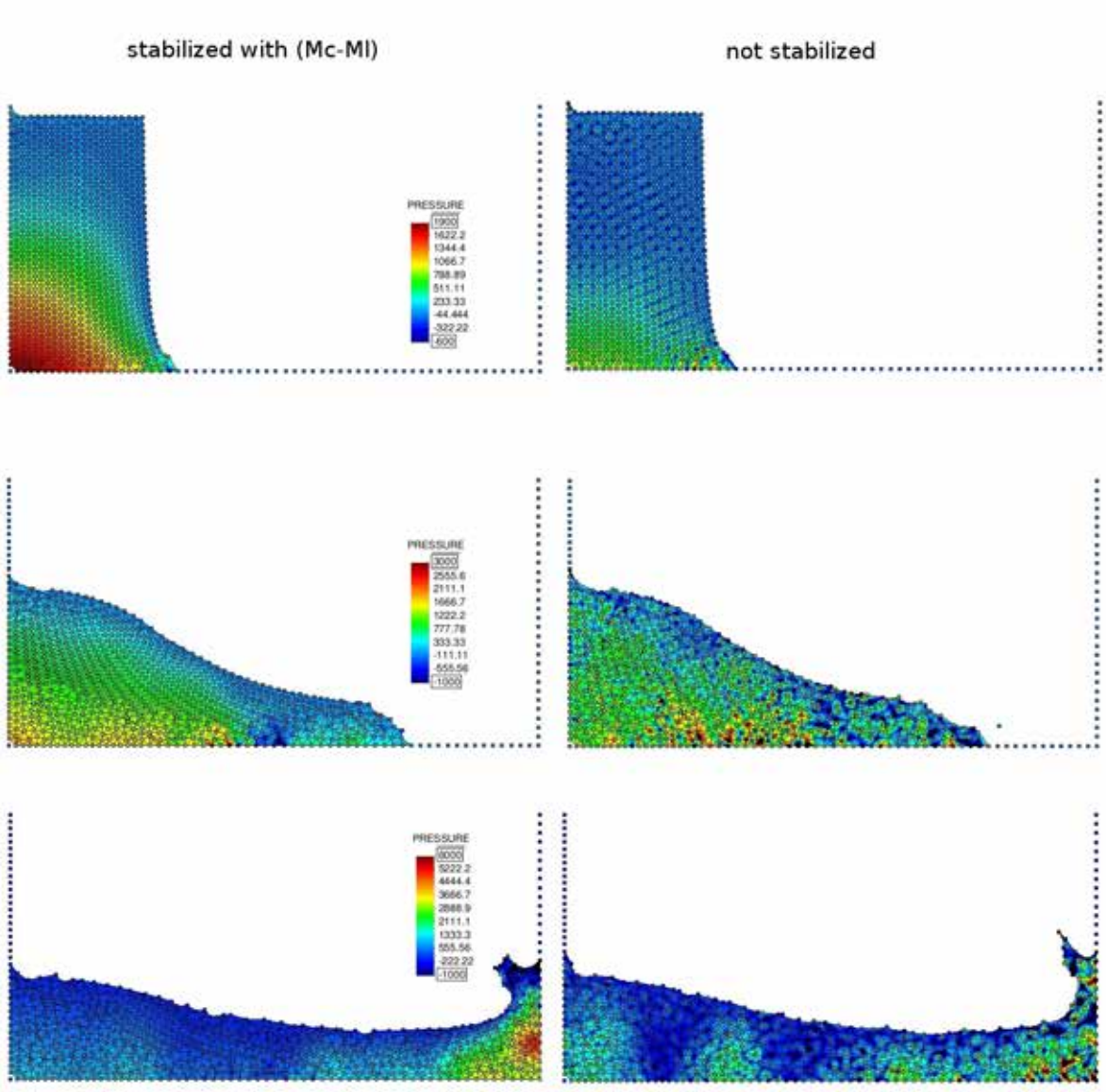


Figure 3.6: Dam break. Pressure stability.

Results obtained with quasi-incompressible updated Lagrangian fluid formulation (ULF) developed in this chapter are compared with the experimental ones reported in [61]. The comparison is shown in Fig. 3.8. Water was modeled with $\kappa = 10^7$ [Pa]. Snap shots are taken at time instances t=0.04, 0.08, 0.12, 0.16 [s].

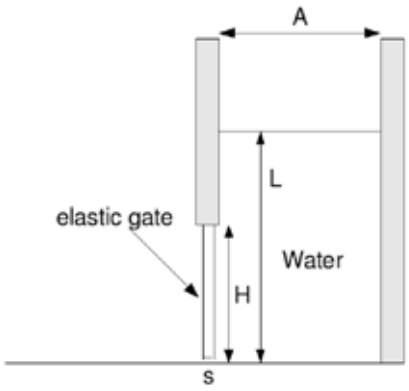


Figure 3.7: Deformation of an elastic plate subjected to water pressure

3.6.3 Floating ball

The first 3D examples solved simulates the motion of the inflated ball within a fluid volume. The example should serve as a qualitative illustration of the applicability of the method. A ball of 0.15 [m] radius is immersed in the middle of the cube with the edge length of 1 [m]. The ball is modeled as a membrane with 0.003 [m] thickness. The properties of the membrane ball are summarized next:

- density $\rho = 800$ $[kg/m^3]$
- Young modulus $E = 0.81$ [MPa]
- Poisson's ratio $\nu = 0.1$

The internal pressure of 15 KPa is applied to the sphere. The total fluid-membrane domain is meshed with 80.000 linear tetrahedra. For the fluid the bulk modulus $K = 10E6$ [Pa] is chosen. Fig. 3.9 illustrates the ball motion. Initially the ball expands, until the equilibrium with the external pressure is achieved and then moves up towards to the surface, the first two figures illustrate a 2D cut and the last two show a 3D view.

3.6.4 Deformation of an elastic balloon filled with water

Next, the results of another 3D simulation are presented. A truncated spherical shape membrane filled with water is fixed at its bottom and then is exposed to the water weight. The membrane has 0.1 [m] radius, truncation base radius of 0.05 [m], 1.5 [mm] thickness and the following physical properties:

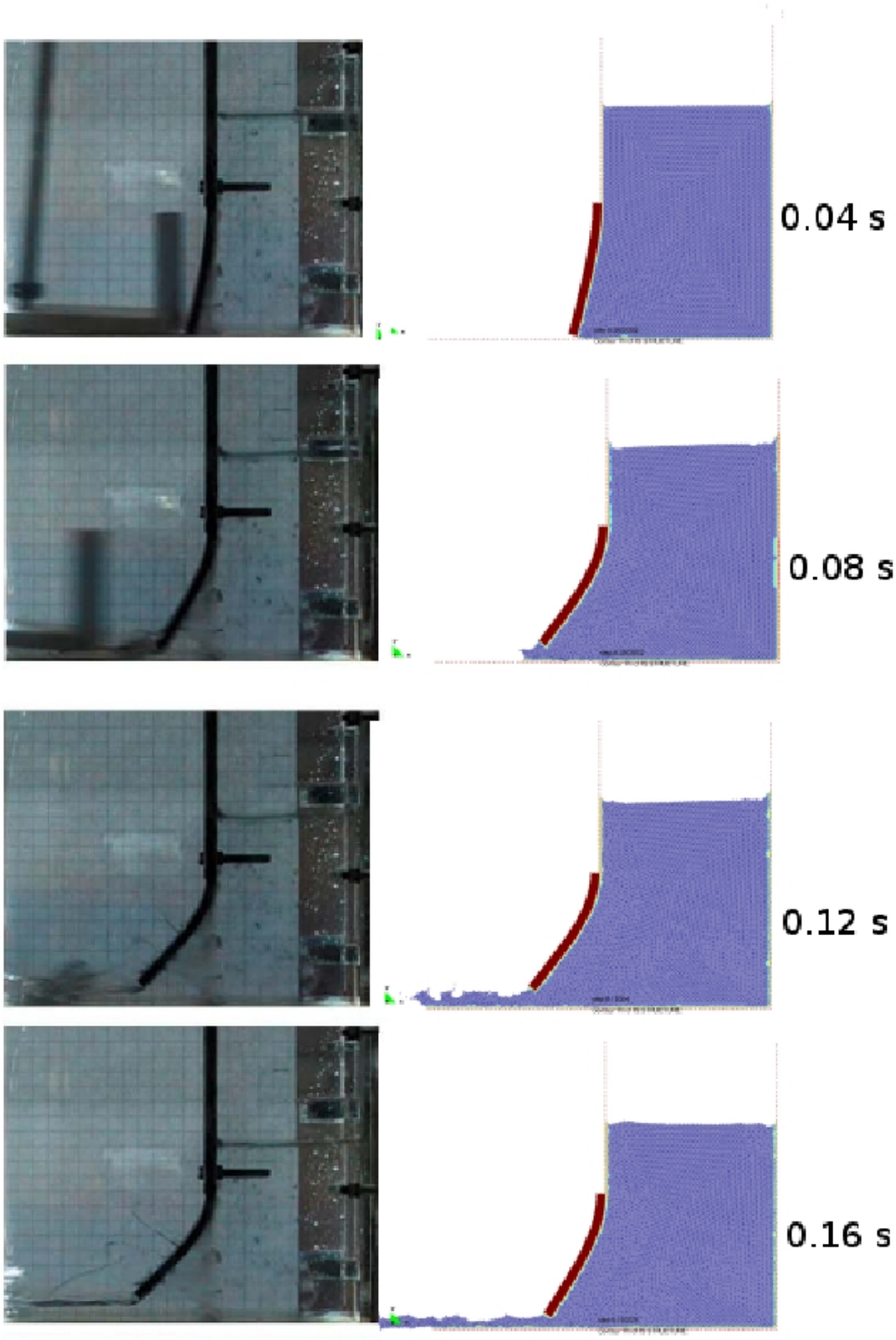
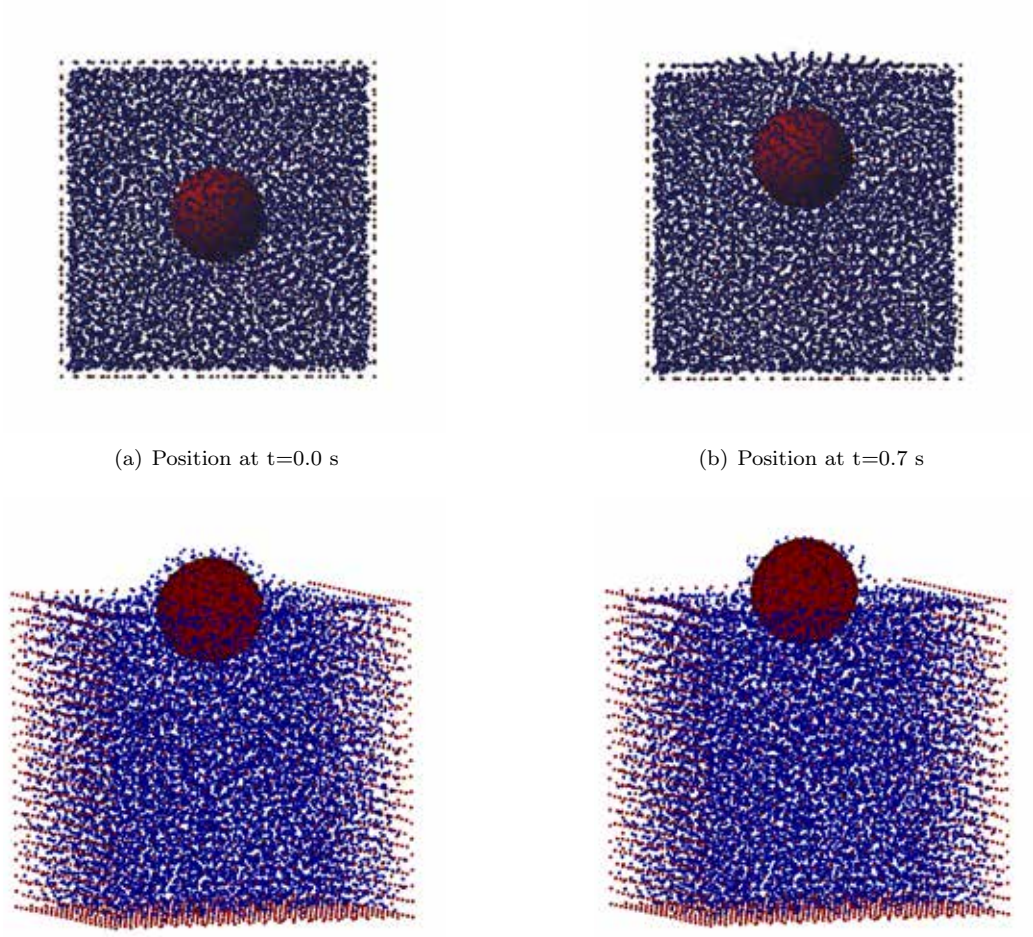


Figure 3.8: Deformation of an elastic plate subjected to water pressure. Comparison with experiment



(a) Position at t=0.0 s

(b) Position at t=0.7 s

(c) Position at t=2.0 s

(d) Position at t=5.0 s

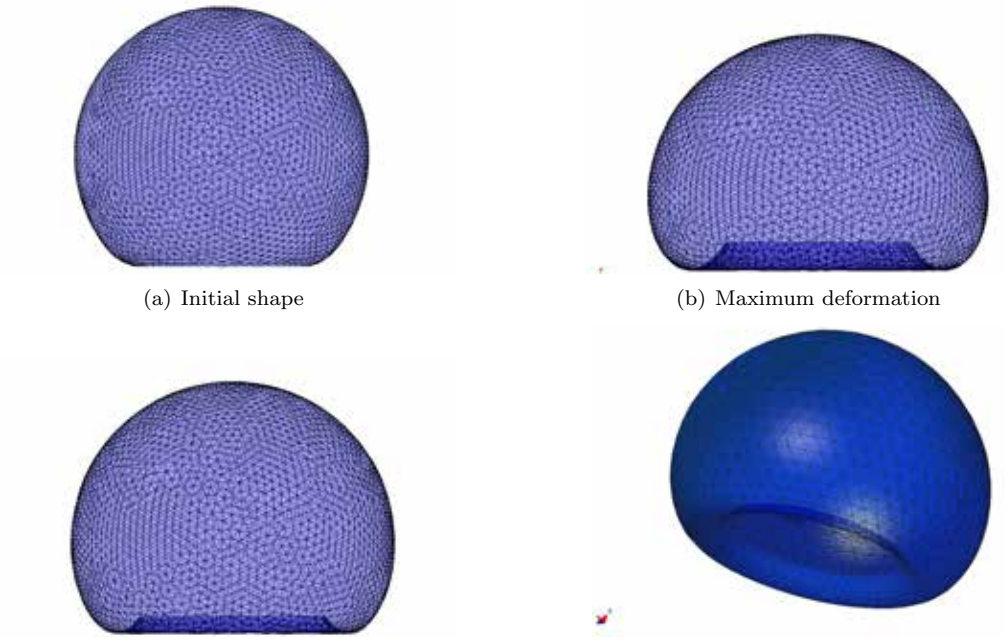
Figure 3.9: Motion of an inflated ball in water

- density $\rho = 1100$ $[kg/m^3]$
- Young modulus $E = 0.8$ [MPa]
- Poisson ratio $\nu = 0.2$

Results are presented for the water bulk modulus $\kappa = 10^6$ [Pa]. Fig. 3.10 depicts the membrane at three time instances: the initial undeformed stage, maximum deformation and the final state.

Fig. 3.11 presents the evolution of the vertical displacement of the upper-most point of the membrane obtained with three different meshes:

1. 3500 fluid linear tetrahedra and 1200 membrane triangles
2. 8.000 fluid linear tetrahedra and 2300 membrane triangles



(a) Initial shape (b) Maximum deformation

(c) Equilibrium state (d) 3D view of maximum deformation

Figure 3.10: Deformation of a membrane filled with water

3. 20000 fluid linear tetrahedra and 4000 membrane triangles

This example belongs to the type for which the superiority of the monolithic Lagrangian method is obvious. The motion of the membrane and its contact with the fluid is tracked automatically when the proposed approach is applied. This example does not require very high values of the bulk modulus, as the velocities encountered are low, thus the only thing that has to be assured for the correct representation of the physical phenomenon is merely minimizing the local and global volume variations. This example reflects the volume variation due to the introduced compressibility and is not associated with the mass losses due to free surface determination using the alpha shape method. Therefore it shows the deviation of the proposed quasi-incompressible FE formulation from the ideally incompressible formulation that theoretically should result in zero mass changes. Table 3.6 shows the maximum mass changes for different values of bulk modulus. The relative volume variation is computed as follows (density is assumed constant): $a_{max} = \frac{\sum_{el} V_{el}^n}{V_0}$, where $V_0 = \sum_{el} V_{el}(t=0)$ is the fluid volume at the initial state, and V_{el}^n is the volume at the current step. The greatest volume loss and the volume loss at the equilibrium state are displayed.

One can see that already for the bulk modulus of K=1 [MPa] the resulting mass loss is less than 0.5 %. This clearly shows that the introduced slight compressibility does not introduce any considerable error as long as the volume conservation is concerned.

κ	a_{max} [%]	a_{end} [%]
10E05	2.43	1.6
10E06	0.26	0.14
10E07	0.015	0.01

Table 3.6: Maximal and average volume variation: Example 3.6.4

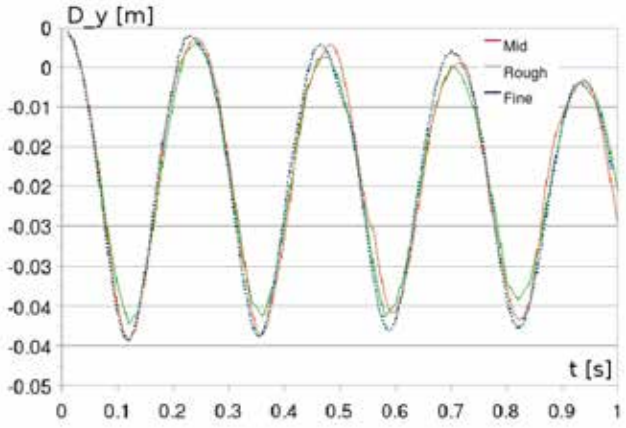


Figure 3.11: Comparison of the vertical displacement for the membrane filled with water obtained with three different meshes

3.7 Summary and conclusions

In this chapter a new Lagrangian fluid formulation adopting the general philosophy of PFEM was developed. It was shown that the fluid formulation can be naturally coupled with any displacement-based Lagrangian solid into a FSI monolithic method. The improved discrete system conditioning was achieved by condensing the pressure. Pressure condensation required modeling the fluid as slightly compressible, choosing adequate values of the compressibility constant. However, in contrast with alternative methods where elemental pressure condensation was used due to discontinuous pressure elements, we proposed the use of the linear displacement-pressure elements. Thus, the pressure is condensed at a global level. The use of continuous pressure interpolation allowed us to obtain a locking-free formulation. It was shown that the computational effort associated with the global pressure condensation can be minimized by using a matrix-free iterative method for the solution of the linearized system. The simple pressure stabilization technique introduced does not preclude pressure condensation due to the lumped pressure mass matrix multiplying the unknown end-of-step pressure.

It was shown that the non-linear loop convergence of the fluid solver does not deteriorate sig-

nificantly when the structure is included. For sufficiently small time step values the convergence is achieved in 3-4 iterations.

Despite of its efficiency for moderate values of the bulk modulus, it was discovered that the convergence of the iterative linear solver deteriorates with the increase of the bulk modulus as the discrete system becomes "stiff". However, for the values of the bulk modulus up to $\approx 10E6$ the convergence speed makes the method competitive and practically useful. In particular, it was shown that the volume alterations at such bulk modulus are less than 1% of the total mass. The great advantage of the method is the fact that the numerical solution does not underestimate the displacement/velocity field for any value of the bulk modulus even on rough meshes.

The limitation of the method originates from the deteriorating convergence rates of the iterative solver used here. A possibility of using a more sophisticated linear solver and/or pre-conditioner defines an open line of research.

Chapter 4

A solver for sub-sonic thermally coupled flows in an Eulerian framework

4.1 Introduction and outline

The title of this chapter deviates at the first glance from the topic of this work. However, the developments of this chapter serve the following purpose: they are used as one of the ingredients of the solver presented in the next chapter (**Chapter 5**), where Lagrangian and Eulerian formulations will be combined to obtain an efficient solution algorithm of an involved practical problem.

On the other hand, the developments of the current chapter are valuable by themselves as they present a method for the solution of thermally coupled flow problems in the sub-sonic regime. The formulation proposed here can be used for solving a wide range of problems dealing with low-speed (sub-sonic) flows. We restrict ourselves here to flows in enclosures, an area where Eulerian methods are the most efficient.

Additionally, this chapter permits us to view in the classical techniques as well as the difficulties encountered when the Eulerian formulation is adopted and thus have a clear understanding of the differences with the Lagrangian approach. In particular, we address here such important issues as stabilization of convection and the fractional step technique in the presence of convection.

The chapter is organized as follows: first, the general problems arising when modeling thermally

coupled flows are highlighted. Classification according to the Mach number of the flow is introduced. The governing equation system of a compressible flow is presented. The differences between the conservation and non-conservation forms of the equations are addressed. The impact of using either form in different flow regimes is explained. Next we concentrate on the area of interest, namely low speed flows. First, the simplest model, i.e. the one based on Boussinesq hypothesis is derived. Details with respect to the discretization, time integration and stabilization are presented. A combination of 4th order Runge-Kutta time integration scheme with the fractional step split is introduced. The Galerkin/least squares (GLS) technique for convection and pressure stabilization is applied.

Following a similar strategy the fully compressible formulation is derived via the non-conservation form of the governing equations. We present a method that allows splitting the original problem consisting of coupled energy(convection-diffusion)/momentum/continuity/state equations into a set of sub-problems. We show that a simple way of uncoupling the energy equation from the rest of the system relies on obtaining the temperature from the energy equation treated explicitly. Next we propose a semi-explicit method for integrating the momentum-continuity system in time, based on a combination of Runge-Kutta and Backward Euler schemes. Finally the fractional-step equations for the compressible case are derived. The chapter concludes with benchmark examples validating the method.

4.2 Low speed compressible flows

The description of compressible flows is complicated as in the most general case it requires simultaneous consideration of both the mechanical and thermodynamic aspects of the fluid. In comparison with the incompressible case, the necessity of tracking a new variable, namely the density in the continuity equation, requires definition of an equation of state to complete the system. This equation in general case is of the type $\rho = f(p, T)$. As a consequence the energy equation is coupled to the rest of the governing equations. The resulting system of equations turns out to be strongly coupled and highly non-linear.

Governing equations A compressible flow is described by a coupled system of continuity-momentum-energy equations equipped with the equation of state which is necessary for completing the system.

For a viscous compressible fluid the equations are [9]:

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} - \boldsymbol{\sigma}) = \rho \mathbf{b} \tag{4.1}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{4.2}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\mathbf{v} (\rho E + p)) = \rho g \cdot \mathbf{v} + Q + \Psi + \nabla \cdot q \tag{4.3}$$

$$\rho = f(p, T) \tag{4.4}$$

where the variables are velocity $\mathbf{v}$, pressure p , density ρ, total energy $E = e + \rho \frac{\mathbf{v} \cdot \mathbf{v}}{2}$ (the sum of internal and kinetic energies), $\mathbf{q}$ is the heat flux vector, $\mathbf{Q}$ and Ψ the internal heat generation and internal heat dissipation functions respectively, $\boldsymbol{\sigma}$ is the Cauchy stress tensor. The viscous dissipation function for a Newtonian fluid is defined as

$$\Psi = \nabla \cdot (2\mu \tilde{\epsilon} (\mathbf{v})) \tag{4.5}$$

where and $\tilde{\epsilon}$ is the deviatoric strain rate. The equations are written in the so-called "conservation form", which refers to an arrangement of an equation, that shows that a property represented is conserved by making its overall change equal to zero.

Mach number It is convenient to introduce the following classification of compressible flows on the basis of a dimensionless parameter, the Mach number defined as

$$Ma = \frac{||\mathbf{v}||}{c} \tag{4.6}$$

where $\mathbf{v}$ and c are the velocity of the flow and the sound velocity in the considered medium.

Based upon the Mach number value, flows can be classified as

1. low Mach flow ($Ma << 1$)
2. subsonic flow ($Ma < 1$)
3. trans-sonic flow ($0.8 < Ma < 1.2$)
4. supersonic flow ($1.2 < Ma < 3$)
5. hyper-sonic flow ($Ma > 3$)

As we shall see, compressible flows exhibit distinct physical behavior depending on the flow regimes. In each of such regimes different physical phenomena are dominating, which generally

speaking precludes the efficient modeling of a general compressible flow using a single numerical model. At qualitative level several new phenomena are present in compressible flows in contrast to the formerly discussed incompressible ones:

- presence of sound waves
- possibility to develop shocks

Sound The sound is a travelling wave which is an oscillation of pressure (deviation from the mean) transmitted through the medium. Sound waves are composed of frequencies within the hearing range thus giving the name to this type of waves. The speed of sound c can be calculated as

$$c = \sqrt{\frac{K}{\rho}} \tag{4.7}$$

where K is the bulk modulus. For example the speed of sound in the air at normal conditions is about $c_{air} \approx 340$ [m/s]. For incompressible flows the speed of sound is infinite, and the pressure fluctuations transmit instantaneously in the flow. In principle, the presence of sound waves in compressible flows does not present any qualitative complication. However in some flow regimes (in particular in low Mach number flows) sound waves introduce oscillations at a scale very different from that of the bulk flow. Therefore, if the goal of the analysis is the description of the overall flow phenomenon, it may be appealing to filter the sound out either already at the continuous level (by modifying the governing equations) or by numerical damping [62]. In any case, when designing a solution scheme for a low Mach number flow one must assure that the time scale related to the sound does not affect the time step size of the time integration.

Shock waves In the sub-sonic regime, the information propagates faster than the moving object/flow, thus permitting the slow or still regions of the flow to adjust continuously to the arriving disturbance. On the other hand, when an object moves with a speed greater than the speed of sound (super-sonic regime), the medium in the upstream direction does not obtain the information upon its arrival prior to immediate contact, thus leading to presence of locations where the high speed and low speed coexist. For an inviscid flow the two solutions lead to a purely discontinuous behaviour. In case of a viscous flow they are separated by a thin viscous layer. At the boundary of the sub-sonic and super-sonic regions a shock occurs, which is an abrupt, nearly discontinuous, change in the characteristics of the medium. Across a shock there is always an extremely rapid variation in pressure, temperature and density of the flow.

Mathematically shocks are an intrinsic feature of the Euler equations [9] and do not exist (in the precise sense) for the viscous case. However the steepness of the "shock" is defined by the width of the viscous layer, thus for low viscosity (e.g. in the case of air), practically this viscous width stands much below resolvable length-scales. Therefore, from a practical point of view shocks are also present in the Navier-Stokes equations. It can be shown mathematically that the shocks appear in the Euler equations when all eigen-values of the diagonalization are positive, which turns out to happen for $Ma \geq 1$ [63].

The main analytic difficulty associated to the presence of the shocks is that the solutions of even the strong statement of the problem are not anymore standard differentiable functions. However, in a weak formulation it is possible to move the differential operator to the test function (by using partial integration), thus permitting the non-differentiability of the solution[1]. In turn this means that in the presence of shocks one is obliged to use the conservation form of the governing equations, since the transformation into the non-conservation strong form is valid only under the assumption of the differentiability of the unknown (already at the continuous level). It is shown that the solutions of the non-conservation form for $Ma \geq 1$ do not satisfy the second law of thermodynamics [9], thus explaining why the non-conservation form gives the wrong position and intensity of the shocks, i.e. non-physical results.

Low Mach number regime The nature of low Mach and high Mach number flows are very different. For isothermal conditions low Mach number flows behave as incompressible. For example an isothermal flow at $Ma = 0.3$ leads to a density change of $\approx 5\%$ (see e.g. [64]). Thus for low Mach number flows the constant density assumption is often acceptable from the physical point of view.

However if thermal effects are not negligible, the situation becomes qualitatively different. The density becomes a function of both pressure and temperature ($\rho = f(p, T)$) thus requiring to take into account the energy equation.

Boussinesq approximation For small temperature gradients, one usually implies the so-called Boussinesq approximation. This approximation consists in neglecting density changes of the flow except for those that are multiplying the buoyancy term.

The set of equations solved is equivalent to that of the incompressible case. The continuity equation is the incompressible one ($\nabla \cdot \mathbf{u} = 0$) and therefore the energy equation remains uncoupled. In fact the formulation remains incompressible and thus the changes necessary to implement the

[1] E.g. let us assume that the function ∇f is our solution that contains a discontinuity, however multiplying it with a test function $(q, \nabla f) = (\nabla q, f)$

Boussinesq approximation are minimal. Practically the correction factor for the gravity is calculated as

$$\mathbf{g}^{new} = \mathbf{g}_s(1 - \alpha(T_{new} - T_s)) \tag{4.8}$$

where α is the Boussinesq coefficient and index "s" stands for standard conditions.

Even though the Boussinesq model is very popular, the bound of its application is the permissible temperature alterations for which the Boussinesq hypothesis holds and in the correct prediction of the thermal flow expansion. Unfortunately, quantitative results that provide such bounds do not exist yet. Gartling and Hickox [65] presented results that provide quantitative limits for the Boussinesq approximation in terms of assumed tolerance for satisfaction of the incompressibility constraint. This however does not set a criterion that directly links the value of the temperature change with the accuracy in terms of primary variables (velocity, temperature). The conclusion is that one can use Boussinesq hypothesis only in cases when the prediction of the correct density distribution is of no importance, and when the temperature gradients appearing in the problem remain qualitatively small. In practice, one may use Boussinesq approximation when the nondimensionalized temperature differences encountered in the problem ($\frac{\Delta T}{T_{av}}$) are less than $\propto 10\%$.

For low speed flows in the presence of large temperature gradients and thermally induced density alteration the Boussinesq approximation has to be discarded. The type of problems we are aiming to resolve here falls into this category. In such case one stands in front of two main choices, namely the "low Mach number model" and a fully compressible sub-sonic model. Both options have a number of advantages and disadvantages and we shall address those here.

Low Mach number equations In low Mach number flows acoustic waves introduce a phenomenon associated with a scale greatly differing from that of the bulk flow. This can be easily illustrated: let us assume that a flow moves with a speed $\mathbf{v}$ and the characteristic length of the problem is L. This will result in a characteristic time $t_1 \propto L/u$. The acoustic pressure waves on the contrary move with the speed of sound c, and the associated characteristic time will be $t_2 \propto L/c$. For low Mach number as we approach $Ma \to 0$, c is tending towards infinity and thus the characteristic time for the acoustic wave is much smaller than that of the bulk flow, $t_2 << t_1$. The phenomena occurring at such small time instances are of no interest for the analysis of the bulk flow and may be even disregarded as fluctuations or "noise". Its removal out of the Navier-Stokes equations is the goal of low Mach number models.

A low Mach number model is obtained by modifying the fully compressible Navier-Stokes equa-

tions so that the acoustic pressure is filtered out already at the continuous level by rescaling the velocity field. This is done by means of first nondimensionalazing the governing equations (see e.g. [5] then expanding the flow variables into power series, and finally neglecting the terms multiplied by high order of Mach number. This is of course justified, when the Mach number of the flow is sufficiently low ($Ma << 1$). For details on low Mach number models, reader is referred to [5], [62].

An important result of the low Mach number model is that the pressure becomes split into a reference thermodynamic pressure and mechanical pressure. The first one is constant over the whole domain and changes its value only by global heating or mass addition. The mechanical pressure component is determined from a velocity constraint, playing the same role as in the incompressible equations.

Low Mach number equations are also widely used as for most thermal convection problems the propagation speed of sound is much greater than the propagation speed of the phenomenon one wishes to study.

Full compressible Navier-Stokes equations for sub-sonic flow The more general approach for modeling sub-sonic flows consists in using the full set of compressible Navier-Stokes equations under certain simplifications that hold just for the sub-sonic regime. In particular using the fact that shock waves are not present in sub-sonic flows it is possible to use the non-conservation form of the governing equations. This is appealing due to several practical reasons. First of all, most incompressible flow solvers are based on the non-conservation form of the equations. Thus changing the incompressible code into a sub-sonic compressible one is quite straight-forward. Second, the momentum equations in the non-conservation form assume the format of a transport equation, which is thoroughly studied and for which rigorous analysis and stabilization techniques are well-established.

For these reasons in the present work we choose the non-conservation form of the governing equations. Next we illustrate how it is obtained from the conservation form and show which steps in this transformation hold only in the absence of shocks, i.e. in the sub-sonic regime. For the sake of clarity we use the indicial form when transforming the momentum equation. First we carry out the partial integration of the term $\nabla \cdot \mathbf{v} \otimes \mathbf{v}$ (we use Einsteins' notation convention with respect to repeated indices. Symbols ∂_i stand for a spatial partial derivative with respect to the i component and ∂_t for the time derivative):

$$\nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) \,|_i = \partial_j \left(\rho \mathbf{v}_j \mathbf{v}_i\right) = \rho \mathbf{v}_j \partial_j \mathbf{v}_i + \mathbf{v}_i \left(\partial_j \left(\rho \mathbf{v}_j\right)\right) \tag{4.9}$$

Note that this partial differentiation is valid **only** if the argument $(\rho \mathbf{v}_j \mathbf{v}_i)$ is differentiable, an assumption holding exclusively for sub-sonic flows, where functions do not exhibit discontinuities.

Similarly we use the product rule for the inertia terms:

$$\partial_t (\rho \mathbf{v}_i) = \rho \partial_t \mathbf{v}_i + \mathbf{v}_i \partial_t \rho \tag{4.10}$$

The sum of the expanded convective and the inertia term give

$$\partial_t (\rho \mathbf{v}_i) + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) \,|_i = \rho \partial_t \mathbf{v}_i + \mathbf{v}_i \partial_t \rho + \rho \mathbf{v}_j \partial_j \mathbf{v}_i + \mathbf{v}_i (\partial_j (\rho \mathbf{v}_j)) = \rho \partial_t \mathbf{v}_i + \rho \mathbf{v}_j \partial_j \mathbf{v}_i + \mathbf{v}_i [\partial_t \rho + \partial_j (\rho \mathbf{v}_j)] \tag{4.11}$$

Noting that the term in parenthesis is nothing but the continuity equation (we reproduce it here in indicial notation for the sake of clarity)

$$\partial_t \rho + \partial_j (\rho \mathbf{v}_j) = 0 \tag{4.12}$$

and substituting it into Eq. (4.11) results in

$$\partial_t (\rho \mathbf{v}_i) + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) \,|_i = \rho \mathbf{v}_j \partial_j \mathbf{v}_i + \rho \partial_t \mathbf{v}_i \tag{4.13}$$

Adding the rest of the terms (which are equivalent in conservation and non-conservation forms) we obtain

$$\rho \mathbf{v}_j \partial_j \mathbf{v}_i + \rho \partial_t \mathbf{v}_i + \partial_i p - \partial_j (\mu (\partial_j \mathbf{v}_i + \partial_i \mathbf{v}_k)) = \rho \mathbf{b} \tag{4.14}$$

which is the non-conservation form of the momentum equation. Next we write the Eq. (4.14) in the symbolic notation

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} + \nabla p - \nabla (2\mu \tilde{\epsilon}(\mathbf{v})) = \rho \mathbf{b} \tag{4.15}$$

The derivation of the non-conservation form of the energy equation is slightly more involved. For the sake of clarity we reproduce here the energy equation (4.3)

$$\frac{D}{Dt} \left[\rho \left(\frac{\mathbf{v} \cdot \mathbf{v}}{2} + e \right) \right] + \rho \left(\frac{\mathbf{v} \cdot \mathbf{v}}{2} + e \right) \nabla \cdot \mathbf{v} = \nabla \cdot (k \nabla T) + \nabla \cdot (\mathbf{v} \cdot \boldsymbol{\sigma}) + \mathbf{v} \cdot \rho \mathbf{b} + Q \tag{4.16}$$

We start by expanding the total derivative of the kinetic energy using the product rule, which is possible if $\rho \mathbf{v} \cdot \mathbf{v}$ is differentiable. This is assured only in the absence of shocks, i.e. the sub-sonic

case.

$$\frac{D}{Dt}\left(\rho\frac{\mathbf{v}\cdot\mathbf{v}}{2}\right) = \frac{1}{2}\rho\frac{D(\mathbf{v}\cdot\mathbf{v})}{Dt} + \frac{\mathbf{v}\cdot\mathbf{v}}{2}\frac{D\rho}{Dt} \tag{4.17}$$

Next we use the continuity equation to express the temporal derivative of density as

$$\frac{1}{2}\rho\frac{D(\mathbf{v}\cdot\mathbf{v})}{Dt} + \frac{\mathbf{v}\cdot\mathbf{v}}{2}\frac{D\rho}{Dt} = \rho\mathbf{v}\frac{D\mathbf{v}}{Dt} - \frac{\mathbf{v}\cdot\mathbf{v}}{2}\rho\nabla\cdot\mathbf{v} \tag{4.18}$$

Having done this, we can rewrite the conservation of energy noting that the product of the kinetic energy and the divergence of velocity cancels out, i.e.

$$\rho\mathbf{v}\frac{D\mathbf{v}}{Dt} + \frac{D\rho e}{Dt} + \rho e\nabla\cdot\mathbf{v} = \nabla\cdot(\mathbf{v}\cdot\boldsymbol{\sigma}) + \mathbf{v}\cdot\rho\mathbf{b} + Q - \frac{\partial q}{\partial x} \tag{4.19}$$

Next we subtract the momentum equation multiplied by velocity ($\rho\mathbf{v}\cdot\frac{D\mathbf{v}}{Dt} = \mathbf{v}\cdot\rho\mathbf{b} + \mathbf{v}\cdot\nabla\cdot\boldsymbol{\sigma}$) from Eq. (4.19), and noting that the terms $\rho\mathbf{v}\cdot\mathbf{b}$ and $\rho\mathbf{v}\frac{D\mathbf{v}}{Dt}$ cancel out results in

$$\frac{D\rho e}{Dt} + \rho e\nabla\cdot\mathbf{v} = \nabla\cdot(\mathbf{v}\cdot\boldsymbol{\sigma}) - \mathbf{v}\cdot\nabla\cdot\boldsymbol{\sigma} + Q - \frac{\partial q}{\partial x} \tag{4.20}$$

Finally, using $\frac{D\rho e}{Dt} = \rho\frac{De}{Dt} - e\rho\nabla\cdot\mathbf{v}$ and $\nabla\cdot(\mathbf{v}\cdot\boldsymbol{\sigma}) = \nabla\mathbf{v} : \boldsymbol{\sigma} + \mathbf{v}\cdot\nabla\cdot\boldsymbol{\sigma}$ yields

$$\rho\frac{De}{Dt} = \nabla\mathbf{v} : \boldsymbol{\sigma} + Q - \frac{\partial q}{\partial x} \tag{4.21}$$

Using the constitutive equations for the stress $\boldsymbol{\sigma} = -pI + 2\mu\tilde{\epsilon}(\mathbf{v})$ (where $\tilde{\epsilon}$ is the deviatoric rate of strain tensor) and the diffusive heat flux ($q = -k\nabla T$) for a Newtonian fluid we obtain :

$$\rho\frac{De}{Dt} = -p\nabla\cdot\mathbf{v} + 2\mu\tilde{\epsilon}(\mathbf{v}) : \boldsymbol{\sigma} - \nabla\cdot(k\nabla T) + Q \tag{4.22}$$

The term $\Psi = 2\mu\epsilon : \boldsymbol{\sigma}$ corresponds to the mechanical energy dissipation due to shear motion. This term is usually neglected for flows with low viscosity and at low speeds, which is justified in our case (for the justification based upon consideration of the coefficients in the nondimensionalized form of the energy equation see e.g. [66]). However, for the sake of generality we leave it in the equation.

For an ideal gas the internal energy increment is related to the temperature as

$$\delta e = c_v\delta T \tag{4.23}$$

where c_v is the isochoric heat capacity. This permits us to simplify Eq. (4.22) as

$$\rho c_v \frac{DT}{Dt} + p\nabla \cdot \mathbf{v} = \nabla \cdot (k\nabla T) + \Psi + Q \tag{4.24}$$

which is the non-conservation form of the energy equation for an ideal gas.

The full governing equation system thus be written as

$$\rho \frac{\partial \mathbf{v}}{\partial t} - \nabla \cdot \boldsymbol{\sigma} + \rho (\mathbf{v} \cdot \nabla) \mathbf{v} = \rho \mathbf{b} \tag{4.25}$$

$$\frac{\partial \rho}{\partial t} + \rho \nabla \cdot \mathbf{v} + \mathbf{v} \cdot \nabla \rho = 0 \tag{4.26}$$

$$\rho c_v \frac{DT}{Dt} + p\nabla \cdot \mathbf{v} = \nabla \cdot (k\nabla T) + \Psi + Q \tag{4.27}$$

$$\rho = \frac{p}{RT} \tag{4.28}$$

Above equations shall serve as the starting point for the following models: Boussinesq model (under further simplifications), and a fully compressible sub-sonic model.

4.3 The FE model based on the Boussinesq assumption

This section we present the simplest way of modeling thermally coupled flows at low speeds, via the Boussinesq model.

Derivation of the discrete equations for the Boussinesq model The basis of the Boussinesq hypothesis are the incompressible Navier-Stokes equations. The assumption of accounting for the density only in the gravity term further simplifies the equation system (4.25-4.28). Assuming otherwise constant density we obtain the incompressible form of the continuity equation $\nabla \cdot \mathbf{v} = 0$. Plugging this into Eq. (4.27) we obtain the well-known form of the convection-diffusion equation for the incompressible fluid:

$$\rho c_v \frac{DT}{Dt} = \nabla \cdot (k\nabla T) + Q + \Psi \tag{4.29}$$

The momentum equation (4.25) simplifies to

$$\rho \frac{\partial \mathbf{v}}{\partial t} - \nabla \cdot \boldsymbol{\sigma} + \rho (\mathbf{v} \cdot \nabla) \mathbf{v} = \rho \mathbf{b}^{new} \tag{4.30}$$

with the Boussinesq term $\mathbf{b}^{new}$ for the gravity according to Eq. (4.8).

The presence of the Boussinesq gravity term in the momentum equation generally couples the

convection-diffusion and momentum/continuity system. However it is standard to solve them in a sequential way. Note that in a standard incompressible case (i.e. without the modified gravity term), momentum/continuity system does not depend on the solution of the convection-diffusion equation, thus leaving the coupling of the mentioned equation exclusively due to the velocity entering the convective term of the convection-diffusion equation.

FE model Next we shall derive the FE model. This is organized as follows: we start by deriving the discrete form of the incompressible version of the equations. Stabilization, time integration issues and fractional step split will be addressed. Having done this, we will introduce the modification of the gravity term according to the Boussinesq assumption (4.8) and then present the solution strategy.

Stability As we have seen in **Chapters 2** and **3** the Lagrangian incompressible fluid formulation required exclusively pressure stabilization terms to be added to the Galerkin weak form of the problem. In the Eulerian formulation the situation changes due to the presence of the convective term. It turns out that if the viscosity ν is small (convection dominated flows) the control of the velocity gradients becomes very weak and in particular the convective term comes out of control. This can be understood by noting that the momentum equations behave as a transport equation that needs to be stabilized for high Peclet numbers [9].

From the practical point of view, purely Galerkin-based solutions of convection-dominated flows turn out to exhibit severe oscillations or "wiggles" in the velocity field. This happens because the convective operator is not symmetric. If the viscosity of the flow is large, the associated discrete operator (Laplacian) dominates and the solution becomes stable[2].

Pressure stability was addressed in some detail in the previous chapter. We will just remind here that the basic idea of the pressure stabilization consists in enforcing the positive definiteness of the system matrix for inf-sup instable (e.g. standard linear velocity-pressure) elements.

Even though in the course of history methods have been proposed to heal the two instabilities mentioned (convection and inf-sup) separately, modern stabilization methods (Galerkin/least squares, Algebraic Sub-scales, Orthogonal Sub-scales, Finite Calculus) stabilize pressure and convection together. The idea of most of stabilization methods consists in modifying the weak form of the problem by adding an additional term, that is typically an L_2 product of the residual of the equation with an operator applied to the test function finally multiplied by an algorithmic param-

[2]The mathematical explanation of this phenomenon is based upon the so-called Cea's lemma, which leads to a conclusion that the error bound of Galerkin method is inversely proportional to the diffusion (viscosity in case of Navier-Stokes) coefficient. Thus for low values of viscosity the control is lost [9].

eter (these stabilization methods are sometimes termed "residual-based"). Quoting Codina [67] *the different methods differ in which is the operator acting on the test function and in the design of the algorithmic parameter.*

Galerkin/ least squares (GLS) stabilization The incompressible Navier-Stokes equations accounting for thermal changes of buoyancy in the Eulerian framework can be written as

$$\rho\frac{\partial \mathbf{v}}{\partial t} - \rho\nu\Delta\mathbf{v} + \rho(\mathbf{v}\cdot\nabla)\mathbf{v} + \nabla p = \tilde{\mathbf{f}} \tag{4.31}$$

$$\nabla\cdot\mathbf{v} = 0 \tag{4.32}$$

equipped with the Dirichlet boundary conditions (note that here $\tilde{\mathbf{f}} = \rho\mathbf{b}^{new}$, according to the Boussinesq hypothesis).

Let us define the Navier-Stokes operator:

$$\mathcal{L}(\mathbf{v}, p) = \begin{pmatrix} \mathcal{L}_1(\mathbf{v}, p) \\ \mathcal{L}_2(\mathbf{v}, p) \end{pmatrix} \tag{4.33}$$

where

$$\mathcal{L}_1 = -\rho\nu\Delta\mathbf{v} + \nabla p + \rho(\mathbf{v}\cdot\nabla)\mathbf{v} \tag{4.34}$$

and

$$\mathcal{L}_2 = \nabla\cdot\mathbf{v} \tag{4.35}$$

Then introducing the notation $\mathcal{F} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}$, the incompressible Navier-Stokes equations can be written as

$$\begin{pmatrix} \rho\frac{\partial \mathbf{v}}{\partial t} \\ 0 \end{pmatrix} + \mathcal{L}(\mathbf{v}, p) = \mathcal{F} \tag{4.36}$$

The Galerkin weak form of (4.36) reads

$$\left(\mathbf{w}, \rho\frac{\partial \mathbf{v}}{\partial t}\right) + (\mathbf{w}, \mathcal{L}_1) + (q, \mathcal{L}_2) = \langle \mathbf{F}, \mathbf{w}\rangle \tag{4.37}$$

where $\mathbf{w}$ and q are the velocity and pressure test functions. (We follow the standard notation, where $(\cdot,\cdot)$ means a bilinear form and $\langle\cdot,\cdot\rangle$ - the linear form). Note that the discrete momentum and continuity equations were summed up.

The stabilization consists in adding the term $\sum_n \int_\Omega \mathcal{L}(\mathbf{w}, q) \cdot (\mathcal{F} - \mathcal{L}(\mathbf{v}, p) - \frac{\partial \mathbf{v}}{\partial t})$ to the Galerkin weak form [9]. Stabilization is achieved by adding the product of residual with Navier-Stokes operator $\mathcal{L}(\mathbf{w}, q)$ which can be seen as a modified weight function (instead of the standard weight function $\mathbf{w}$). It is clear that such term does not violate the consistency of the formulation - as the solution is approached the residual becomes zero and we recover the solution of the original equation.

Adding the stabilization terms to the Galerkin weak form leads to

$$(\mathbf{w}, \frac{\partial \mathbf{v}}{\partial t}) + (\nabla \cdot \mathbf{v}, p) - \nu(\mathbf{w}, \Delta \mathbf{v}) + (\mathbf{w}, \mathbf{v} \cdot \nabla \mathbf{v}) + (\nabla \cdot \mathbf{v}, q) + \sum_{el} \int_{\Omega_{el}} \tau \left(\nabla q - \nu \Delta \mathbf{w} + \mathbf{w} \cdot \nabla \mathbf{w}\right) \left(\mathbf{F} - \frac{\partial \mathbf{v}}{\partial t} - \nabla p + \nu \Delta \mathbf{v} - \mathbf{v} \cdot \nabla \mathbf{v}\right) \tag{4.38}$$

In Eq. (4.38) terms that are multiplied by the velocity test function $\mathbf{w}$ are momentum and by the pressure test function q - the continuity equations contributions. Rearranging the terms, performing the integration by parts of the viscous term, and noting that for the linear velocity-pressure elements the Laplacian operator $\Delta = 0$, one obtains modified momentum and continuity equations:

$$(\mathbf{w}, \frac{\partial \mathbf{v}}{\partial t}) + (\nabla \cdot \mathbf{v}, p) - \nu(\nabla \mathbf{w}, \nabla \mathbf{v}) + (\mathbf{w}, \mathbf{v} \cdot \nabla \mathbf{v}) + \sum_{el} \int_{\Omega_{el}} \tau \left(\mathbf{w} \cdot \nabla \mathbf{w}\right) \left(\mathbf{F} - \frac{\partial \mathbf{v}}{\partial t} - \nabla q - \mathbf{v} \cdot \nabla \mathbf{v}\right) = \langle \mathbf{F}, \mathbf{w} \rangle \tag{4.39}$$

$$(q, \nabla \cdot \mathbf{v}) + \sum_{el} \int_{\Omega_{el}} \tau \nabla q \left(\mathbf{F} - \frac{\partial \mathbf{v}}{\partial t} - \nabla p - \mathbf{v} \cdot \nabla \mathbf{v}\right) \tag{4.40}$$

with the algorithmic stabilization parameter τ defined according to [54] as

$$\tau = \frac{2||\mathbf{v}||}{h} + \frac{4\nu}{h^2} \tag{4.41}$$

with $\mathbf{v}$ being convective velocity, ν the kinematic viscosity and h the element size.

Remark One of the methods that were developed more recently is the so-called Algebraic Sub-grid Scale Method (ASGS). For a general case it is claimed to introduce less artificial diffusion than the GLS method. However GLS and ASGS methods lead to the same stabilized discrete equations for the linear elements. This originates from the fact that in comparison with the GLS, the modified test function of ASGS is based on the adjoint operator $\mathcal{L}^*$, which for Navier-Stokes problem is defined

[2]Adjoints of operators generalize conjugate transposes of square matrices to infinite-dimensional situations. If one thinks of operators on a Hilbert space as "generalized complex numbers", then the adjoint of an operator plays the role of the complex conjugate of a complex number. Formally, given an inner product space $\mathcal{H}$, with the conjugate symmetry property $\langle f, g \rangle = \overline{\langle g, f \rangle}$ and an operator T, the conjugate T^* is an operator such that $\langle f, Tg \rangle = \langle T^* f, g \rangle$

as follows.

$$\mathcal{L}^* = \begin{pmatrix} \rho\nu\Delta\mathbf{w} + \nabla q + \rho(\mathbf{v}\cdot\nabla)\mathbf{v} \\ \nabla\cdot\mathbf{v} \end{pmatrix} \tag{4.42}$$

Comparing $\mathcal{L}^*$ and $\mathcal{L}$ one can see that the only difference consists in the sign of the viscous term. This term is zero for linear elements, and therefore GLS and ASGS coincide for such case.

Before writing the discrete form of the stabilized equations we shall define the discrete operators that correspond to the terms entering the Galerkin part of the weak form are defined as follows

$$(\mathbf{w},\mathbf{v}) \longrightarrow \mathbf{M}\bar{\mathbf{v}} \tag{4.43}$$

$$(\nabla\cdot\mathbf{w},p) \longrightarrow \mathbf{G}\bar{p} \tag{4.44}$$

$$(\nabla\mathbf{w},\nabla\mathbf{v}) \longrightarrow \mathbf{L}\bar{\mathbf{v}} \tag{4.45}$$

$$(\mathbf{w},(\mathbf{v}\cdot\nabla\mathbf{v})) \longrightarrow \mathbf{K}(\bar{\mathbf{v}})\bar{\mathbf{v}} \tag{4.46}$$

$$\langle\mathbf{f},\mathbf{w}\rangle \longrightarrow \mathbf{F} \tag{4.47}$$

and the stabilization terms:

$$\sum_{el}\int_{\Omega_{el}} \tau(\mathbf{w}\cdot\nabla\mathbf{w})(\mathbf{w}\cdot\nabla\mathbf{v}) \longrightarrow \mathbf{S}_1\bar{\mathbf{v}} \tag{4.48}$$

$$\sum_{el}\int_{\Omega_{el}} \tau(\mathbf{w}\cdot\nabla\mathbf{w})(\mathbf{v}) \longrightarrow \mathbf{S}_2\bar{\mathbf{v}} \tag{4.49}$$

$$\sum_{el}\int_{\Omega_{el}} \tau(\mathbf{w}\cdot\nabla\mathbf{w})(\nabla p) \longrightarrow \mathbf{S}_3\bar{p} \tag{4.50}$$

$$\sum_{el}\int_{\Omega_{el}} \tau(\mathbf{w}\cdot\nabla\mathbf{w})(\frac{\partial\mathbf{v}}{\partial t}) \longrightarrow \mathbf{M}_{st}\frac{\partial\bar{\mathbf{v}}}{\partial t} \tag{4.51}$$

$$\sum_{el}\int_{\Omega_{el}} \tau(\mathbf{w}\cdot\nabla\mathbf{w})(\mathbf{f}) \longrightarrow \mathbf{F}_{s1} \tag{4.52}$$

$$\sum_{el}\int_{\Omega_{el}} \tau(\nabla q)(\mathbf{f}) \longrightarrow \mathbf{F}_{s2} \tag{4.53}$$

$$\sum_{el}\int_{\Omega_{el}} \tau(\nabla q)(\mathbf{v}) \longrightarrow \mathbf{S}_4\bar{\mathbf{v}} \tag{4.54}$$

for all $f, g \in \mathcal{H}$ [68].

$$\sum_{el} \int_{\Omega_{el}} \tau(\nabla q)((\mathbf{w} \cdot \nabla)\mathbf{v}) \longrightarrow \mathbf{S}_5 \bar{\mathbf{v}} \tag{4.55}$$

$$\sum_{el} \int_{\Omega_{el}} \tau(\nabla q)(\nabla p) \longrightarrow \mathbf{L}_\tau \bar{p} \tag{4.56}$$

Now we can write the matrix form of the stabilized semi-discrete equations:

$$(\mathbf{M} + \mathbf{M}_{st}) \frac{\partial \bar{\mathbf{v}}}{\partial t} - (\mu \mathbf{L} + \mathbf{K}(\bar{\mathbf{v}}) + \mathbf{S}_1 + \mathbf{S}_2)\bar{\mathbf{v}} + (\mathbf{S}_3 + \mathbf{G})\bar{p}) = \mathbf{F} + \mathbf{F}_{s1} \tag{4.57}$$

$$(\mathbf{D} + \mathbf{S}_4 + \mathbf{S}_5)\bar{\mathbf{v}} + \mathbf{L}_\tau \bar{p} = \mathbf{F}_{s2} \tag{4.58}$$

Since the explicit time integration is applied, the mass matrix multiplying the acceleration is used in the lumped format. As the stabilization term $\mathbf{M}_{st}$ is also multiplying the unknown acceleration $\bar{\mathbf{a}}_{n+1}$, it would be necessary to lump it as well for using in the explicit time integration context. Another option consists in approximating it by $\mathbf{M}\bar{\mathbf{a}}_n$, thus adding it only to the right-hand side. This option is followed here:

$$\mathbf{M}_{st} \frac{\partial \bar{\mathbf{v}}}{\partial t} = \mathbf{M}_{st} \bar{\mathbf{a}}_{n+1} \approx \mathbf{M}_{st} \bar{\mathbf{a}}_n \tag{4.59}$$

Let us introduce the auxiliary notation:

$$\bar{\mathbf{K}} = \mu \mathbf{L} + \mathbf{K}(\mathbf{v}_h) + \mathbf{S}_1 + \mathbf{S}_2 \tag{4.60}$$

$$\bar{\mathbf{G}} = \mathbf{S}_3 + \mathbf{G} \tag{4.61}$$

$$\bar{\mathbf{F}} = \mathbf{F} + \mathbf{F}_{s1} + \mathbf{M}_{st} \bar{\mathbf{a}}_n \tag{4.62}$$

$$\bar{\mathbf{D}} = \mathbf{D} + \mathbf{S}_4 + \mathbf{S}_5 \tag{4.63}$$

$$\hat{\mathbf{F}} = \mathbf{F}_{s2} \tag{4.64}$$

This notation will serve us for carrying out the subsequent time integration without getting confused by the numerous stabilization matrices. Using the introduced short-hand notation, the semi-discrete equations read

$$\mathbf{M} \frac{\partial \bar{\mathbf{v}}}{\partial t} + \bar{\mathbf{K}} \bar{\mathbf{v}} + \bar{\mathbf{G}} \bar{p} = \bar{\mathbf{F}} \tag{4.65}$$

$$\bar{\mathbf{D}} \bar{\mathbf{v}} + \mathbf{L}_\tau \bar{p} = \hat{\mathbf{F}} \tag{4.66}$$

Having introduced this notation we proceed with the time integration.

Time integration Generally explicit time integration techniques are faster as they neither involve non-linear loops, nor the solution of a linear system (in case the mass matrix lumping is used). Their disadvantage consists in conditional stability and therefore the necessity of time step estimation. However in the incompressible regime, time step estimates are favorable. The simplicity of their implementation also contributes to their popularity. The simplest explicit scheme, namely the Forward Euler method, consists in evaluating all the residual terms at the preceding time step. That is for a general Cauchy problem

$$\frac{\partial y}{\partial t} = f(y(t), t) \tag{4.67}$$

the Forward Euler scheme gives

$$\frac{y_{n+1} - y_n}{\Delta t} = f(y_n, t_n) \tag{4.68}$$

Unfortunately this scheme is only first order accurate. A very powerful class of explicit methods is the Runge-Kutta multi-step family. Runge-Kutta schemes consist in evaluating the residual at a number of intermediate steps. It is well known that the n-step Runge-Kutta schemes give accuracy of nth order up to $n = 4$. For $n > 4$ the order is less than n. This makes the 4-step method particularly popular. Also with respect to stability properties (critical time step), the 4th order Runge-Kutta scheme defines the optimal compromise between the number of intermediate steps and the permissible time step size (with respect to conditional stability of the explicit scheme). For details of the stability domain of Runge-Kutta schemes reader is referred to **Section 5.3.3.2** of [9].

For the incompressible Navier-Stokes problem a fully explicit time scheme cannot be used due to implicit nature of pressure. Therefore we propose to use an explicit 4th order Runge-Kutta algorithm for time integration of the momentum equation, while treating the pressure term implicitly.

4th order Runge-Kutta scheme Given the solution at t_n and looking for the solution at t_{n+1} the Runge-Kutta algorithm involves subsequent evaluation of the equation residual at a few intermediate times instances. For a general Cauchy problem of type Eq. (4.67) the scheme gives the following formula:

$$\frac{y_{n+1} - y_n}{\Delta t} = \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4] \tag{4.69}$$

where $k_1 ... k_4$ are the residuals (or in other words the "right-hand side" corrections) that are computed according to the formulae:

$$k_1 = f(t_n, y_n) \tag{4.70}$$

$$k_2 = f(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2} k_1) \tag{4.71}$$

$$k_3 = f(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2} k_2) \tag{4.72}$$

$$k_4 = f(t_n + \Delta t, y_n + \Delta t k_3) \tag{4.73}$$

This means that the computation of the y_{n+1} requires evaluating the right-hand side four times and updating the unknown at each of the intermediate steps.

Now we shall illustrate the application of this scheme to our problem, namely the integration of the momentum equation. Let us define the residual of the momentum equation as

$$\mathbf{r} = \bar{\mathbf{F}} - \bar{\mathbf{K}}\bar{\mathbf{v}} - \bar{\mathbf{G}}\bar{p} \tag{4.74}$$

In doing so we slightly violate our notation, as $\mathbf{r}$ is defined at each node and should have been indicated as $\bar{\mathbf{r}}$. We will need, however, to introduce later the symbol $\bar{\mathbf{r}}$ and $\hat{\mathbf{r}}$ for the modified residua. The we prefer to leave the initial $\mathbf{r}$ without the overbar.

Then the sequence of residuals necessary for carrying out the integration according to the Eqs. (4.70-4.73) reads:

$$\mathbf{r}_1 = \bar{\mathbf{F}}_n - \bar{\mathbf{K}}(\bar{\mathbf{v}}_n)(\bar{\mathbf{v}}_n) - \bar{\mathbf{G}}\bar{p}_{\beta 1} \tag{4.75}$$

$$\bar{\mathbf{v}}_{\beta 1} = \bar{\mathbf{v}}_n + \mathbf{M}^{-1}\frac{\Delta t}{2}\mathbf{r}_1 \tag{4.76}$$

$$\mathbf{r}_2 = \bar{\mathbf{F}}_{\beta 1} - \bar{\mathbf{K}}(\bar{\mathbf{v}}_{\beta 1})(\bar{\mathbf{v}}_{\beta 1}) - \bar{\mathbf{G}}\bar{p}_{\beta 2} \tag{4.77}$$

$$\bar{\mathbf{v}}_{\beta 2} = \bar{\mathbf{v}}_n + \mathbf{M}^{-1}\frac{\Delta t}{2}\mathbf{r}_2 \tag{4.78}$$

$$\mathbf{r}_3 = \bar{\mathbf{F}}_{\beta 2} - \bar{\mathbf{K}}(\bar{\mathbf{v}}_{\beta 2})(\bar{\mathbf{v}}_{\beta 2}) - \bar{\mathbf{G}}\bar{p}_{\beta 3} \tag{4.79}$$

$$\bar{\mathbf{v}}_{\beta 3} = \bar{\mathbf{v}}_n + \mathbf{M}^{-1}\Delta t\mathbf{r}_3 \tag{4.80}$$

$$\mathbf{r}_4 = \bar{\mathbf{F}}_{n+1} - \bar{\mathbf{K}}(\bar{\mathbf{v}}_{\beta 3})(\bar{\mathbf{v}}_{\beta 3}) - \bar{\mathbf{G}}\bar{p}_{\beta 4} \tag{4.81}$$

Note that we assume that the external force (gravity) $\bar{\mathbf{F}}$ varies due to the Boussinesq modification of gravity term.

Computation of the right-hand-side corrections $r_1, ..., r_4$ exactly would require the solution of

the continuity equation for the pressure at each intermediate step in order to assure that each of the intermediate velocities are divergence-free. Doing so we would assure the 4th order accuracy of the resulting scheme. In practice we propose not to do it as it would considerably slow down the method, since solving the continuity equation involves a linear system solution (that cannot be avoided as the Laplacian operator cannot be lumped). Instead we assume the linear variation of pressure and the gravity force, which permits us to write: $\bar{\mathbf{F}}_{\beta 1} = \bar{\mathbf{F}}_{\beta 2} = \bar{\mathbf{F}}(t_n + \frac{\Delta t}{2}) = \frac{1}{2}\left(\bar{\mathbf{F}}_n + \bar{\mathbf{F}}_{n+1}\right)$

- $\bar{p}_{\beta 1} = \bar{p}(t_n) = \bar{p}_n$
- $\bar{p}_{\beta 2} = \bar{p}(t_n + \frac{\Delta t}{2}) = \frac{1}{2}(\bar{p}_n + \bar{p}_{n+1})$
- $\bar{p}_{\beta 3} = \bar{p}(t_n + \frac{\Delta t}{2}) = \frac{1}{2}(\bar{p}_n + \bar{p}_{n+1})$
- $\bar{p}_{\beta 4} = \bar{p}(t_{n+1}) = \bar{p}_{n+1}$

Remark Note that not solving the continuity equation at every intermediate step of Runge-Kutta scheme, but rather once, implies that only the final velocity is divergence-free[3]. This does not introduce any error in the context of a fixed-grid approach as the one here, but may violate the geometric conservation law thus leading to problems when mesh-moving techniques are used [69].

Following (4.69), we obtain the discrete momentum equation.

$$\bar{\mathbf{M}}\frac{\bar{\mathbf{v}}_{n+1} - \bar{\mathbf{v}}_n}{\Delta t} = \frac{1}{6}[\mathbf{r}_1 + 2\mathbf{r}_2 + 2\mathbf{r}_3 + \mathbf{r}_4] \tag{4.82}$$

The introduced assumption (linearity of pressure between t_n and t_{n+1}) allows us to take the pressure gradient terms out of the residual, giving a form convenient for the further application of the fractional step method:

$$\bar{\mathbf{M}}\frac{\bar{\mathbf{v}}_{n+1} - \bar{\mathbf{v}}_n}{\Delta t} = \frac{1}{6}[\bar{\mathbf{r}}_1 + 2\bar{\mathbf{r}}_2 + 2\bar{\mathbf{r}}_3 + \bar{\mathbf{r}}_4] + \frac{1}{2}\bar{\mathbf{G}}\left(\bar{p}_{n+1} + \bar{p}_n\right) \tag{4.83}$$

[3]Linear one-step methods such as Backward Euler or Crank-Nicholson methods assure that if $\mathbf{D}\mathbf{v}_{n+1} = 0$ and $\mathbf{D}\mathbf{v}_n = 0$, then $\mathbf{D}\mathbf{v} = 0$ for any $t_n \leq t \leq t_{n+1}$. To assure this in the case of quadratic method such as BDF2 three time instances are necessary t_{n-1}, t_n and t_{n+1}. In comparison with these schemes, Runge-Kutta is based upon several subsequent re-evaluations of the residual and not on polynomial assumption for the integrand function. Thus does not assure that if $\mathbf{D}\mathbf{v}_{n+1} = 0$ and $\mathbf{D}\mathbf{v}_n = 0$, then for any time instant between t_n and t_{n+1} the velocity is divergence-free during the entire step. However, ensuring that the continuity equation is satisfied in the whole time interval is important only if there are non-potential forces present, since they are the ones that depend on the path rather than on the initial and final state. In our case, where either air or water are the objects of modeling, the viscosity is sufficiently low.

The modified residuals are defined as (we indicate the modified residua by an overbar):

$$\bar{\mathbf{r}}_1 = \bar{\mathbf{F}}_n - \bar{\mathbf{K}}(\bar{\mathbf{v}}_n)(\bar{\mathbf{v}}_n) \tag{4.84}$$

$$\bar{\mathbf{r}}_2 = \frac{1}{2}\left(\bar{\mathbf{F}}_n + \bar{\mathbf{F}}_{n+1}\right) - \bar{\mathbf{K}}(\bar{\mathbf{v}}_{\beta 1})(\bar{\mathbf{v}}_{\beta 1}) \tag{4.85}$$

$$\bar{\mathbf{r}}_3 = \frac{1}{2}\left(\bar{\mathbf{F}}_n + \bar{\mathbf{F}}_{n+1}\right) - \bar{\mathbf{K}}(\bar{\mathbf{v}}_{\beta 2})(\bar{\mathbf{v}}_{\beta 2}) \tag{4.86}$$

$$\bar{\mathbf{r}}_4 = \bar{\mathbf{F}}_{n+1} - \bar{\mathbf{K}}(\bar{\mathbf{v}}_{\beta 3})(\bar{\mathbf{v}}_{\beta 3}) \tag{4.87}$$

$$\tag{4.88}$$

thus implying that $\mathbf{r} = \bar{\mathbf{r}} + \mathbf{G}\bar{p}$

Fractional Step Method The system of equations to be solved consists of the momentum equation (4.83) and the continuity equation, whose final form reads:

$$\bar{\mathbf{D}}\bar{\mathbf{v}}_{n+1} + \mathbf{L}_\tau \bar{p}_{n+1} = \hat{\mathbf{F}} \tag{4.89}$$

In order to split the solution for velocity and pressure, thus permitting us to solve the momentum equation four times (that is implied by evaluation of the right-hand side corrections r_i for 4th order Runge-Kutta method) while solving the continuity equation only once, we shall apply the fractional step split (explained in the **Chapter 2**) by introducing intermediate, non-solenoidal velocity $\tilde{\mathbf{v}}$.

The fractional split applied to (4.83) leads to

$$\mathbf{M}\frac{\tilde{\mathbf{v}} - \bar{\mathbf{v}}^n}{\Delta t} = \frac{1}{6}[\bar{\mathbf{r}}_1 + 2\bar{\mathbf{r}}_2 + 2\bar{\mathbf{r}}_3 + \bar{\mathbf{r}}_4] + \gamma\bar{\mathbf{G}}\bar{p}_n \tag{4.90}$$

$$\mathbf{M}\frac{\bar{\mathbf{v}}_{n+1} - \tilde{\mathbf{v}}}{\Delta t} + \frac{1}{2}\bar{\mathbf{G}}(\bar{p}_{n+1} - \gamma\bar{p}_n) = 0 \tag{4.91}$$

$$\bar{\mathbf{D}}\bar{\mathbf{v}}_{n+1} + \mathbf{L}_\tau \bar{p}_{n+1} = \hat{\mathbf{F}} \tag{4.92}$$

(the $\frac{1}{2}$ coefficient multiplying the pressure gradients in the end-of-step are due to the linear pressure

assumption for evaluation of the pressure at intermediate steps of the Runge-Kutta scheme) with

$$\tilde{\mathbf{r}}_1 = \bar{\mathbf{F}}_n - \bar{\mathbf{K}}(\bar{\mathbf{v}}_n)(\bar{\mathbf{v}}_n) \tag{4.93}$$

$$\tilde{\mathbf{r}}_2 = \frac{1}{2}\left(\bar{\mathbf{F}}_n + \bar{\mathbf{F}}_{n+1}\right) - \bar{\mathbf{K}}(\tilde{\bar{\mathbf{v}}}_{\beta 1})(\tilde{\bar{\mathbf{v}}}_{\beta 1}) \tag{4.94}$$

$$\tilde{\mathbf{r}}_3 = \frac{1}{2}\left(\bar{\mathbf{F}}_n + \bar{\mathbf{F}}_{n+1}\right) - \bar{\mathbf{K}}(\tilde{\bar{\mathbf{v}}}_{\beta 2})(\tilde{\bar{\mathbf{v}}}_{\beta 2}) \tag{4.95}$$

$$\tilde{\mathbf{r}}_4 = \bar{\mathbf{F}}_{n+1} - \bar{\mathbf{K}}(\tilde{\bar{\mathbf{v}}}_{\beta 3})(\tilde{\bar{\mathbf{v}}}_{\beta 3}) \tag{4.96}$$

$$\tag{4.97}$$

Note that now the residuals of the fractional momentum equations correspond to the fractional velocities and not the end-of-step velocity.

For higher accuracy [29] a second order fractional step method is implemented here, thus choosing $\gamma = 1$. The splitting error introduced by the fractional step is due to the approximation of the "generalized" convective operator $\bar{\mathbf{K}}(\mathbf{v}_{n+1}) \approx \bar{\mathbf{K}}(\tilde{\mathbf{v}})$ that enters the residuals[4]. That means that we assume that the convective terms are evaluated with the non-solenoidal velocity and are not corrected in the final step. The error is of second order in time for $\gamma = 1$ [29].

Applying the continuity equation (4.89) to Eq. (4.91), leads to

$$\mathbf{M}\frac{\tilde{\mathbf{v}} - \bar{\mathbf{v}}^n}{\Delta t} = \frac{1}{6}[\tilde{\mathbf{r}}_1 + 2\tilde{\mathbf{r}}_2 + 2\tilde{\mathbf{r}}_3 + \tilde{\mathbf{r}}_4] + \bar{\mathbf{G}}\bar{p}_n \tag{4.98}$$

$$\mathbf{M}\frac{\bar{\mathbf{v}}^{n+1} - \tilde{\mathbf{v}}}{\Delta t} + \frac{1}{2}\bar{\mathbf{G}}(\bar{p}_{n+1} - \bar{p}_n) = 0 \tag{4.99}$$

$$\frac{1}{2}\Delta t \mathbf{D}\mathbf{M}^{-1}\mathbf{G}(\bar{p}_{n+1} - \bar{p}_n) = \bar{\mathbf{D}}\bar{\mathbf{v}} - \hat{\mathbf{F}} - \mathbf{L}_\tau \bar{p}_{n+1} \tag{4.100}$$

Next we introduce the usual approximation for the Laplacian operator $\mathbf{D}\mathbf{M}^{-1}\mathbf{G} \approx \mathbf{L}$ and ultimately arrive at

$$\mathbf{M}\frac{\tilde{\mathbf{v}} - \bar{\mathbf{v}}^n}{\Delta t} = \frac{1}{6}[\tilde{\mathbf{r}}_1 + 2\tilde{\mathbf{r}}_2 + 2\tilde{\mathbf{r}}_3 + \tilde{\mathbf{r}}_4] + \bar{\mathbf{G}}\bar{p}_n \tag{4.101}$$

$$\frac{\Delta t}{2}\mathbf{L}(\bar{p}_{n+1} - \bar{p}_n) = \bar{\mathbf{D}}\bar{\mathbf{v}} - \hat{\mathbf{F}} - \mathbf{L}_\tau \bar{p}_{n+1} \tag{4.102}$$

$$\mathbf{M}\frac{\bar{\mathbf{v}}^{n+1} - \tilde{\mathbf{v}}}{\Delta t} + \frac{1}{2}\bar{\mathbf{G}}(\bar{p}_{n+1} - \bar{p}_n) = 0 \tag{4.103}$$

which defines our discrete model.

[4] In the application of fractional step to a Lagrangian formulation (**Chapter 2**) there was no such error due to the absence of the convective operator.

Algorithm After introducing the scheme for solving the momentum-continuity system the final solution algorithm for the Boussinesq hypothesis is shown in the Table 4.1:

1. solve the convection-diffusion equation and obtain the temperature field $\bar{T}_{n+1}$
2. compute the gravity correction according to the Boussinesq Eq. (4.8).
3. solve the fractional momentum equation (4.101) for $\tilde{\mathbf{v}}$
4. compute the pressure $\bar{p}_{n+1}$ using continuity equation (4.102)
5. solve the end-of-step momentum equation (4.103) for $\bar{\mathbf{v}}_{n+1}$
6. go to next time step

Table 4.1: Algorithm for the thermally coupled flow simulation using Boussinesq assumption

As we see, the Boussinesq approach uses independent modules, namely the thermal and mechanical solver. Dependence of the convective term of the convection-diffusion equation upon the velocity (solution of the momentum/continuity system) and the dependence of gravity term according to Boussinesq upon the temperature couple the two. However the sequential solution of the convection-diffusion and momentum/continuity is usual. The convection-diffusion solver is not specified here as in principle any scheme can be used without affecting the mechanical module. The convection-diffusion equation discretization is addressed in the next section.

Remark 1 The splitting error associated with the fractional step method in the implicit context is due to approximations of the type (e.g. if Crank-Nicholson or Backward Euler scheme is used)

$$K\left(\bar{\mathbf{v}}\left(t_{n+\Theta}\right)\right)\bar{\mathbf{v}}\left(t_{n+\Theta}\right) \approx K\left(\tilde{\mathbf{v}}\right)\tilde{\mathbf{v}} \tag{4.104}$$

A similar error is accepted when a multi-step explicit method such as the Runge-Kutta scheme applied here is used. The error introduced by the split originates from the fact that the intermediate residuals are computed using the intermediate non-divergence free velocities. The approximation can be written as

$$\sum_i \bar{\mathbf{r}}_i(\bar{\mathbf{v}}_{n+1}, \bar{p}_{n+1}) \approx \sum_i \tilde{\mathbf{r}}_i(\tilde{\mathbf{v}}, \bar{p}_n) \tag{4.105}$$

In contrast with the above, when a single-step method that integrates the convection explicitly, e.g. Forward Euler, is used (note that pressure still has to be treated implicitly due to its physical nature) uncoupling of the velocity and pressure can be achieved directly by applying the discrete incompressibility condition $\mathbf{D}\bar{\mathbf{v}}_{n+1}$ to the momentum equation, thus arriving at Poisson's equation for the pressure. In such a case there is no need to use the fractional step technique for decoupling

the velocities and pressure.

The decoupling can be achieved (for simplicity we illustrate it for incompressible inviscid fluid using a Forward Euler time integration scheme for all terms except for the pressure and omit the stabilization terms) by replacing the initial system:

$$\mathbf{M}\frac{\bar{\mathbf{v}}_{n+1}-\bar{\mathbf{v}}_n}{\Delta t}+\mathbf{K}(\bar{\mathbf{v}}_n)+\mathbf{G}\bar{p}_{n+1}=\mathbf{F} \tag{4.106}$$

$$\mathbf{D}\bar{\mathbf{v}}_{n+1}=0 \tag{4.107}$$

by an equivalent one applying the incompressibility condition to the momentum equation, resulting in (note that $\mathbf{D}\bar{\mathbf{v}}_{n+1}=\mathbf{D}\bar{\mathbf{v}}_n=0$)

$$\mathbf{M}\frac{\bar{\mathbf{v}}_{n+1}-\bar{\mathbf{v}}_n}{\Delta t}+\mathbf{K}(\bar{\mathbf{v}}_n)+\mathbf{G}\bar{p}_{n+1}=\mathbf{F} \tag{4.108}$$

$$\mathbf{D}\mathbf{M}^{-1}\mathbf{K}(\bar{\mathbf{v}}_n)+\mathbf{D}\mathbf{M}^{-1}\mathbf{G}\bar{p}_{n+1}=\mathbf{D}\mathbf{M}^{-1}\mathbf{F} \tag{4.109}$$

For an incompressible flow the divergence of gravity is zero, namely $\mathbf{D}\mathbf{M}^{-1}\mathbf{F}=0$ and the discrete Laplacian is again replaced by an approximation (continuous Laplacian) to give finally

$$\mathbf{M}\frac{\bar{\mathbf{v}}_{n+1}-\bar{\mathbf{v}}_n}{\Delta t}+\mathbf{K}(\bar{\mathbf{v}}_n)+\mathbf{G}\bar{p}_{n+1}=\mathbf{F} \tag{4.110}$$

$$\mathbf{D}\mathbf{M}^{-1}\mathbf{K}(\bar{\mathbf{v}}_n)+\mathbf{L}\bar{p}_{n+1}=0 \tag{4.111}$$

The solution is obtained in two simple steps: first the pressure Poisson' equation Eq. (4.111) is solved for $\bar{p}_{n+1}=f(\bar{\mathbf{v}}_n)$. Then the velocity $\bar{\mathbf{v}}_{n+1}=f(\bar{p}_{n+1})$ is computed from Eq. (4.110).

Remark 2 The fractional step method applied to the incompressible Navier-Stokes equations introduces some pressure stability due to the introduction of Laplacian operator. In fact the first order fractional step can be seen as an explicit stabilization in the sense that the Laplacian operator is applied to the pressure at the previous time step n. The time step assumes in such a setting the role of the stabilization parameter. However, the situation changes when for the sake of time accuracy (to achieve second order precision in time) second order fractional step split is used. In such a case the Laplacian is applied to the pressure increment $(\bar{p}_{n+1}-\bar{p}_n)$ and the stabilization effect becomes weaker [29]. Therefore the conventional pressure stabilization techniques are inevitable in this case.

4.4 The FE compressible sub-sonic fluid formulation

The Boussinesq solver derived in the preceding section is limited to small temperature gradients and is unable to predict flow expansion (in the Boussinesq model the thermal influences are accounted for by altering the gravity term, leaving the density constant). In order to treat a wider range of problems we introduce next a fully compressible model, able to describe the whole range of sub-sonic flows.

The starting point are the Navier-Stokes equations in the non-conservation form Eqs. (4.25-4.28), a choice justified by considering only sub-sonic flow regimes, where shock waves are not present and thus the assumption of the differentiability of the solution holds [9], [70]. We strive to develop an approach for sub-sonic flows that can include the incompressible limit as well. Another intention is to preserve the code modularity, that shall be achieved by several system splits (uncoupling the convection-diffusion equation and fractional step for the momentum/continuity) and permit the definition of the thermal and mechanical modules. Code re-usability was also considered, thus the formulation that follows takes as a "departure point" the incompressible code presented in the previous section. Prior to deriving the discrete equations, the choice of the primary variables is addressed.

Variable choice While for the incompressible case it is natural to formulate the governing equations in terms of velocity and pressure, the situation becomes ambivalent in the compressible case. It is appealing, for example, to use the so-called "conservation variables"

$$\mathbf{V}_1^a = \begin{pmatrix} \mathbf{U} \\ \rho \\ E \end{pmatrix} \quad or \quad \mathbf{V}_1^b = \begin{pmatrix} \mathbf{U} \\ p \\ E \end{pmatrix} \tag{4.112}$$

where $\mathbf{U}$ is the momentum, E is the total energy, p is the pressure and ρ is the density, when the equations are written in the conservation form (Eq. 4.1-4.2). In this setting the temporal derivatives are applied directly to the primary variables and not the their products. The set of conservative variables $\mathbf{V}_1^a$are commonly used for high-speed compressible flows [70].

Another set of variables often used in compressible formulations are the so-called "entropy vari-

ables“ defined as

$$\mathbf{V}_2 = \frac{1}{T}\begin{pmatrix} \tilde{\mu} - \frac{u^2}{2} \\ \mathbf{v} \\ -1 \end{pmatrix} \tag{4.113}$$

where μ is the electro-chemical potential per unit mass, $\mathbf{v}$ is the velocity and T is the absolute temperature. The nice property of the resulting discrete system, written in terms of entropy variables, is its symmetry [71] . This property was exploited by Hughes et al. [72] for designing a method for high-speed compressible flows.

Yet another option relies on using the primitive variables, either similarly to the incompressible case

$$\mathbf{V}_3^a = \begin{pmatrix} \mathbf{v} \\ p \\ T \end{pmatrix} \quad or \quad \mathbf{V}_3^b = \begin{pmatrix} \mathbf{v} \\ \rho \\ T \end{pmatrix} \tag{4.114}$$

[5]

It was shown in [73] that only the formulations based on primitive variables ($\mathbf{V}_3^a$)and entropy variables ($\mathbf{V}_2$) give good results in the incompressible limit. The benefit of using primitive variables in contrast with entropy ones consists in its easier implementation within an existing incompressible code (as well as the straight-forward application of existing stabilization techniques)[71]. Thus in the current work the $(p\,, \mathbf{v}\,, T)$ variables set is used.

Having made the choice regarding the primary variables we next concentrate on the time integration of the governing equations. For the sake of clarity we shall omit the stabilization terms and do not discuss here the details of spatial discretization. Discrete matrices follow the definitions of the previous section if not mentioned otherwise.

Residual form of the governing equations In the following we will derive a strategy for the solution of compressible flow problems in the sub-sonic regime. We will work in the residual form and derive first a symbolic linearization of the problem in the form of Newtons iterations.

On the basis of the monolithic system obtained we will derive our approximations step-by-step. The fundamental idea will be to solve the energy equation separately for temperature T to obtain a reduced system of equations that relates $p, \mathbf{v}, \rho$ and finally to use the fractional step approach for the solution of the reduced coupled problem.

[5] Clearly in the incompressible case having density as a primary variable is meaningless. Thus, if it is of interest that the formulation includes the incompressible limit this option must be disregarded.

The introduction of the specific time integration schemes will be left to the very end in order to preserve the generality of the presentation.

The governing equation system consisting of momentum, continuity, energy and state equations (Eqs. (4.25-4.28)) can be written in residual form[6] as (it is convenient to divide the momentum and energy equations by density in order to make the inertial term linear):

$$\begin{pmatrix} \frac{\partial \mathbf{R}_m}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_m}{\partial \rho} & \frac{\partial \mathbf{R}_m}{\partial p} & \frac{\partial \mathbf{R}_m}{\partial T} \\ \frac{\partial \mathbf{R}_c}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_c}{\partial \rho} & \frac{\partial \mathbf{R}_c}{\partial p} & \frac{\partial \mathbf{R}_c}{\partial T} \\ \frac{\partial \mathbf{R}_e}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_e}{\partial \rho} & \frac{\partial \mathbf{R}_e}{\partial p} & \frac{\partial \mathbf{R}_e}{\partial T} \\ \frac{\partial \mathbf{R}_s}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_s}{\partial \rho} & \frac{\partial \mathbf{R}_s}{\partial p} & \frac{\partial \mathbf{R}_s}{\partial T} \end{pmatrix} \begin{pmatrix} d\mathbf{v} \\ d\rho \\ dp \\ dT \end{pmatrix} = \begin{pmatrix} \mathbf{R}_m \\ \mathbf{R}_c \\ \mathbf{R}_e \\ \mathbf{R}_s \end{pmatrix} \tag{4.115}$$

where the momentum R_m, continuity R_c, energy R_e and state R_s equation residuals are defined as

$$\mathbf{R}_m = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} + \frac{1}{\rho} \nabla p - \frac{1}{\rho} \nabla \cdot (2\mu \tilde{\epsilon}(\mathbf{v})) - \mathbf{b} \tag{4.116}$$

$$\mathbf{R}_c = \frac{\partial \rho}{\partial t} + \rho \nabla \cdot \mathbf{v} + \mathbf{v} \cdot \nabla \rho \tag{4.117}$$

$$\mathbf{R}_e = c_v \frac{DT}{Dt} + \frac{p}{\rho} \nabla \cdot u - \frac{1}{\rho} \nabla \cdot (k \nabla T) \tag{4.118}$$

$$\mathbf{R}_s = p - \rho R T \tag{4.119}$$

Observing Eqs. (4.116) and (4.117) we can notice that $\frac{\partial \mathbf{R}_m}{\partial T} = 0$, $\frac{\partial \mathbf{R}_c}{\partial p} = 0$, $\frac{\partial \mathbf{R}_c}{\partial T} = 0$ and $\frac{\partial \mathbf{R}_s}{\partial \mathbf{v}} = 0$. Thus the monolithic system simplifies to

$$\begin{pmatrix} \frac{\partial \mathbf{R}_m}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_m}{\partial \rho} & \frac{\partial \mathbf{R}_m}{\partial p} & 0 \\ \frac{\partial \mathbf{R}_c}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_c}{\partial \rho} & 0 & 0 \\ \frac{\partial \mathbf{R}_e}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_e}{\partial \rho} & \frac{\partial \mathbf{R}_e}{\partial p} & \frac{\partial \mathbf{R}_e}{\partial T} \\ 0 & \frac{\partial \mathbf{R}_s}{\partial \rho} & \frac{\partial \mathbf{R}_s}{\partial p} & \frac{\partial \mathbf{R}_s}{\partial T} \end{pmatrix} \begin{pmatrix} d\mathbf{v} \\ d\rho \\ dp \\ dT \end{pmatrix} = \begin{pmatrix} \mathbf{R}_m \\ \mathbf{R}_c \\ \mathbf{R}_e \\ \mathbf{R}_s \end{pmatrix} \tag{4.120}$$

Note that we have not yet condensed the density via the state equation, but rather left the state equation in the monolithic system. This implies that the "exact" relation between pressure and density is not imposed, and the solution for p and ρ can be obtained separately (contributing to the residual of the state equation).

The system is non-linear and fully coupled (as the off-diagonal terms are non-zero). We can

[6]The residual form is usual for formulating the solution of the monolithic system according to Newtons method. It consists in solving for $d\mathbf{V} = (d\mathbf{v}, dp, d\rho, dT)$, such that $\mathbf{H}d\mathbf{V} - \mathbf{R}(\mathbf{v}_i, p_i, \rho_i, T_i) = 0$, where $\mathbf{H} = \frac{\partial \mathbf{R}}{\partial \mathbf{V}}$ is the tangent matrix. The variable update rule is: $\mathbf{V}_{i+1} = \mathbf{V}_i + d\mathbf{V}$.

clearly see from Eq. (4.120), that in comparison with the incompressible case the energy equation becomes coupled as it contains the terms that depend on pressure, density and velocity, thus generally making all the four terms ($\frac{\partial \mathbf{R}_e}{\partial \mathbf{v}}$, $\frac{\partial \mathbf{R}_e}{\partial \rho}$, $\frac{\partial \mathbf{R}_e}{\partial p}$, $\frac{\partial \mathbf{R}_e}{\partial T}$) in the tangent matrix non-zero.

The solution of the system defined by Eq. (4.120) monolithically is a computationally inefficient procedure. The drawbacks of such approaches due to the presence of different variables in one system were discussed in the previous chapter. Instead we strive to partition this system to obtain a computationally efficient modular approach. Note that up to this point no approximation was introduced and Eq. (4.120) represent the exact linearization of the original set of equations.

Uncoupling the energy equation A simple way of introducing a first split of the system (4.115) (mentioned in [70]) consists in integrating the energy equation explicitly, thus uncoupling it from the rest of the system. In other words we propose to solve the energy equation to obtain T_{n+1}, such that $T_{n+1} = f(p_n, \rho_n, \mathbf{v}_n)$. This means assuming that the partial derivatives with respect to velocity, density and pressure in the left-hand-side of the energy equation vanish: $\frac{\partial T}{\partial \rho} = 0$, $\frac{\partial T}{\partial p} = 0$, $\frac{\partial T}{\partial \mathbf{v}} = 0$. (Note that we have not yet specified the particular type of an explicit time integration scheme to be used.)

Remark This split is automatic if an explicit time integration is assumed, however it can be taken as an approximation if we decide to use an implicit integrator for the temperature.

Having the energy equation decoupled, its solution directly gives the end-of-step temperature that is then used as a known value in the remaining equation, thus the system reduces to

$$\begin{pmatrix} \frac{\partial \mathbf{R}_m}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_m}{\partial \rho} & \frac{\partial \mathbf{R}_m}{\partial p} & 0 \\ \frac{\partial \mathbf{R}_c}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_c}{\partial \rho} & \frac{\partial \mathbf{R}_c}{\partial p} & 0 \\ 0 & 0 & 0 & \frac{\partial \mathbf{R}_e}{\partial T} \\ 0 & \frac{\partial \mathbf{R}_s}{\partial \rho} & \frac{\partial \mathbf{R}_s}{\partial p} & \frac{\partial \mathbf{R}_s}{\partial T} \end{pmatrix} \begin{pmatrix} d\mathbf{v} \\ d\rho \\ dp \\ dT \end{pmatrix} = \begin{pmatrix} \mathbf{R}_m \\ \mathbf{R}_c \\ \mathbf{R}_e \\ \mathbf{R}_s \end{pmatrix} \tag{4.121}$$

and since it is uncoupled can be split into two separate ones that can be solved sequentially, namely

$$\begin{pmatrix} \frac{\partial \mathbf{R}_m}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_m}{\partial \rho} & \frac{\partial \mathbf{R}_m}{\partial p} \\ \frac{\partial \mathbf{R}_c}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_c}{\partial \rho} & \frac{\partial \mathbf{R}_c}{\partial p} \\ 0 & \frac{\partial \mathbf{R}_s}{\partial \rho} & \frac{\partial \mathbf{R}_s}{\partial p} \end{pmatrix} \begin{pmatrix} d\mathbf{v} \\ d\rho \\ dp \end{pmatrix} = \begin{pmatrix} \mathbf{R}_m \\ \mathbf{R}_c \\ \mathbf{R}_s + \left[\frac{\partial \mathbf{R}_s}{\partial T} dT\right]_{=0} \end{pmatrix} \tag{4.122}$$

and

$$\frac{\partial \mathbf{R}_e}{\partial T} dT = \mathbf{R}_e \tag{4.123}$$

(in fact the term in the right-hand-side $\frac{\partial \mathbf{R}_s}{\partial T} dT = 0$ as we assume T to be constant during the iteration (thus dT=0), due to explicit integration of the energy equation).

The next step we shall undertake is the condensation of the density via the state equation and the modification of Eq. (4.122) in order to write the system in terms of exclusively our chosen primary variables $(\mathbf{v}, p, T)$ as unknowns.

Density condensation Using Eq. (4.119) we express the total differential $d\rho$ as

$$d\rho = \frac{1}{RT} dp - \mathbf{R}_s \tag{4.124}$$

The partial differentiation of the state equation residual Eq. (4.119) with respect to $\mathbf{v}, \rho, p$ give

$$\frac{\partial \mathbf{R}_s}{\partial p} = 1 \tag{4.125}$$

$$\frac{\partial \mathbf{R}_s}{\partial \rho} = -RT \tag{4.126}$$

Next we substitute Eq. (4.124) into the system. (4.122), giving

$$\begin{pmatrix} \frac{\partial \mathbf{R}_m}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_m}{\partial \rho} & \frac{\partial \mathbf{R}_m}{\partial p} \\ \frac{\partial \mathbf{R}_c}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_c}{\partial \rho} & \frac{\partial \mathbf{R}_c}{\partial p} \\ 0 & -RT & 1 \end{pmatrix} \begin{pmatrix} d\mathbf{v} \\ \frac{1}{RT}(dp - \mathbf{R}_s) \\ dp \end{pmatrix} = \begin{pmatrix} \mathbf{R}_m \\ \mathbf{R}_c \\ \mathbf{R}_s \end{pmatrix} \tag{4.127}$$

and moving the terms multiplying $\mathbf{R}_s$ to the right-hand side gives finally:

$$\begin{pmatrix} \frac{\partial \mathbf{R}_m}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_m}{\partial \rho} & \frac{\partial \mathbf{R}_m}{\partial p} \\ \frac{\partial \mathbf{R}_c}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_c}{\partial \rho} & \frac{\partial \mathbf{R}_c}{\partial p} \\ 0 & -RT & 1 \end{pmatrix} \begin{pmatrix} d\mathbf{v} \\ \frac{1}{RT} dp \\ dp \end{pmatrix} = \begin{pmatrix} \mathbf{R}_m + \frac{\partial \mathbf{R}_m}{\partial \rho} \frac{\mathbf{R}_s}{RT} \\ \mathbf{R}_c + \frac{\partial \mathbf{R}_c}{\partial \rho} \frac{\mathbf{R}_s}{RT} \\ \mathbf{R}_s + \frac{\partial \mathbf{R}_s}{\partial \rho} \frac{\mathbf{R}_s}{RT} \end{pmatrix} \tag{4.128}$$

Now we plug the partial differentials of the state equation residual Eqs. (4.125) and (4.126) into Eq. (4.128), obtaining (noting that the residual state equation vanishes)

$$\begin{pmatrix} \frac{\partial \mathbf{R}_m}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_m}{\partial \rho} & \frac{\partial \mathbf{R}_m}{\partial p} \\ \frac{\partial \mathbf{R}_c}{\partial \mathbf{v}} & \frac{\partial \mathbf{R}_c}{\partial \rho} & \frac{\partial \mathbf{R}_c}{\partial p} \end{pmatrix} \begin{pmatrix} d\mathbf{v} \\ \frac{1}{RT} dp \\ dp \end{pmatrix} = \begin{pmatrix} \mathbf{R}_m + \frac{\partial \mathbf{R}_m}{\partial \rho} \frac{\mathbf{R}_s}{RT} \\ \mathbf{R}_c + \frac{\partial \mathbf{R}_c}{\partial \rho} \frac{\mathbf{R}_s}{RT} \end{pmatrix} \tag{4.129}$$

Next, grouping together the terms multiplying dp in the left-hand-side we end up with the final

2x2 system that is ready to be discretized:

$$\begin{pmatrix} \frac{\partial \mathbf{R}_m}{\partial \mathbf{v}} & \frac{1}{RT}\frac{\partial \mathbf{R}_m}{\partial \rho} + \frac{\partial \mathbf{R}_m}{\partial p} \\ \frac{\partial \mathbf{R}_c}{\partial \mathbf{v}} & \frac{1}{RT}\frac{\partial \mathbf{R}_c}{\partial \rho} + \frac{\partial \mathbf{R}_c}{\partial p} \end{pmatrix} \begin{pmatrix} d\mathbf{v} \\ dp \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{R}}_m \\ \hat{\mathbf{R}}_c \end{pmatrix} \tag{4.130}$$

where

$$\hat{\mathbf{R}}_m = \mathbf{R}_m + \frac{\partial \mathbf{R}_m}{\partial \rho}\frac{\mathbf{R}_s}{RT} \tag{4.131}$$

$$\hat{\mathbf{R}}_c = \mathbf{R}_c + \frac{\partial \mathbf{R}_c}{\partial \rho}\frac{\mathbf{R}_s}{RT} \tag{4.132}$$

Recall that the temperature in the above equations is considered a known value due to the assumption in Eq. (4.121). The energy equation, completing Eqs. (4.131), (4.132) is repeated here for the sake of clarity:

$$\frac{\partial \mathbf{R}_e}{\partial T}\delta T = \mathbf{R}_e \tag{4.133}$$

Time integration Until this point no time integration scheme was specified. The system defined by Eqs. (4.131-4.133) resembles the one obtained for the incompressible case. This was obtained solely due to the assumption in Eq. (4.121), that allowed the uncoupling of the energy equation from the momentum/continuity system.

Next we proceed with the choice of the time integration schemes, specifying the time integration formulae for the governing equations and obtain the final system, a discrete equivalent of Eqs. (4.130) and Eq. (4.133). At the end we will introduce a fractional step split for Eq. (4.130) in order to solve for velocity and pressure separately.

Energy equation We illustrate the time integration of the energy equation Eq. (4.133) using the simplest explicit scheme, namely Forward Euler. The discrete form of the residual in Eq. (4.117) is

$$\bar{\mathbf{R}}_e = c_v \mathbf{M}\frac{\bar{T}_{n+1} - \bar{T}_n}{\Delta t} - \left(\mathbf{C}_{(\bar{\mathbf{v}}_n)}\bar{T}_n + R\mathbf{D}_{\left(\bar{T}_n\right)}\bar{\mathbf{v}}_n - kL_{(1/\bar{\rho}_n)}\bar{T}_n\right) \tag{4.134}$$

In the above equation $\mathbf{C}$ is the convective operator of a scalar quantity (temperature in this case) that is defined as an assembly of the elemental contributions $\mathbf{C}^e_{IJ} = \int_{\Omega_e} N_I \left(\mathbf{v}\cdot\nabla N_J T_J\right) d\Omega_e$. Additionally we introduce the following notation: given a spatial distribution of a variable $a(x)$, the notation $\mathbf{Z}_{(x)}$ denotes a matrix that is assembled from the elemental contributions multiplied by $a(\mathbf{x}^G)$ evaluated at the Gauss point of an element. For example, having a spatial density distribution $\rho_n(\mathbf{x})$, we write

e.g. $\mathbf{L}_{(1/\rho_n)}$ to denote that $\mathbf{L}_{(1/\rho_n)} = \sum \left(\int_{\Omega_e} \nabla N_I \nabla N_J \frac{1}{\rho_n}^G d\Omega \right)$, where $\frac{1}{\rho}^G$ is the density inverse at a Gauss point. The rest of the matrices here follow the definitions of the previous section.

The operators $\mathbf{C}$, $\mathbf{L}$ and $\mathbf{D}$ are constant as they depend on the values of t_n.

The tangent matrix according to Eq. (4.133) is $\mathbf{H}_e = \frac{\partial \mathbf{R}_e}{\partial T}$:

$$\mathbf{H}_e = \frac{1}{\Delta t}\mathbf{M} \tag{4.135}$$

Momentum/continuity system The solution of the energy equation integrated in time explicitly provides directly the end-of-step temperature distribution , thus making it a known known spatial distribution $T_{n+1}(x)$ for the momentum-continuity system. For the compressible case (in contrast with the incompressible one, where pressure is intrinsically implicit, i.e. it depends only on the current velocity field) the pressure depends on the history. Hence theoretically for time integration of the momentum-continuity system Eq. (4.130) a fully explicit approach can be used. While for values of Mach number ≥ 1, this defines a frequently used option, for low-speed flows it is unfavorable. This relies upon the fact that for low Mach number values, the critical time step is governed by acoustic scales[7] .

A feasible alternative is a semi-explicit scheme, where all the terms are integrated explicitly, except for the pressure, which is integrated implicitly. We propose a combination of the 4th order Runge-Kutta scheme for the momentum equation and Backward Euler scheme for the continuity equation. The high precision and optimality of the Runge-Kutta scheme was addressed in the previous section. Additionally such setting permits re-utilization of parts of the incompressible code, facilitating the implementation. The Backward Euler scheme is chosen due to its dissipative properties, assuring that the spurious frequencies are damped. Thus the proposed setting will contain the acoustic waves that can be correctly resolved by FEM (in contrast e.g. with Low Mach number approach, where no acoustic effects whatsoever are present), and have an efficiency close to an explicit method, while

[7]The stability time step limit of an explicit method is given by the Courant-Friedrichs's law. For an explicit treatment of the pressure terms this gives

$$\Delta t = \frac{CFL}{\frac{||\bar{\mathbf{v}}||}{h} + \frac{c}{h} + \frac{2\mu}{h^2}} \tag{4.136}$$

where CFL is the Courant-Friedrich-Lewy number (usually values between 0.5 and 0.9 are used in practice), $\bar{\mathbf{v}}$ is the bulk flow velocity, c is the sound speed, h is the element size in case FEM is used and μ is the fluid viscosity. For low Mach number flows speed of sound is much larger than the bulk flow speed. The resulting explicit scheme would be useless for predicting the bulk flow behavior having time steps too small for simulating time span that can represent the bulk flow. In contrast, if the pressure is treated implicitly, the Courant-Friedrich-Lewy criterion gives

$$\Delta t = \frac{CFL}{\frac{||\bar{\mathbf{v}}||}{h} + \frac{2\mu}{h^2}} \tag{4.137}$$

and thus only the bulk flow velocity (in case of low fluid viscosity) is governing the time step estimate.

- it has no time step restrictions dictated by the acoustic scales (implicit treatment of pressure)
- it does not contain spurious high frequencies (dissipative time integration)

If second order accuracy in pressure is of interest, one can use e.g. Crank-Nicolson scheme, however the high-frequency damping would be lost[8].

Runge-Kutta scheme for the compressible case The right-hand-side of the momentum equation (Eq. (4.130) and (4.116)) integrated using the Runge-Kutta scheme gives

$$\bar{\mathbf{R}}_m = \mathbf{M}\frac{\bar{\mathbf{v}}_{n+1} - \bar{\mathbf{v}}_n}{\Delta t} - \frac{1}{6}[\mathbf{r}_1 + 2\mathbf{r}_2 + 2\mathbf{r}_3 + \mathbf{r}_4] \tag{4.138}$$

The intermediate residuals[9] are defined as

$$\mathbf{r}_1 = \mathbf{F} - \mathbf{K}(\bar{\mathbf{v}}_n)(\bar{\mathbf{v}}_n) - \mu\mathbf{L}_{(1/\bar{\rho}_n)}\bar{\mathbf{v}}_n + \mathbf{G}_{(1/\bar{\rho}_n)}\bar{p}_n + \\ + \mathbf{G}_{(1/\bar{\rho}_n)}\frac{\bar{p}_n - \bar{\rho}_n R\bar{T}_n}{R\bar{T}_n} \tag{4.139}$$

$$\bar{\mathbf{v}}_{\beta 1} = \bar{\mathbf{v}}_n + \mathbf{M}^{-1}\frac{\Delta t}{2}\mathbf{r}_1 \tag{4.140}$$

$$\mathbf{r}_2 = \mathbf{F} - \mathbf{K}(\bar{\mathbf{v}}_{\beta 1})(\bar{\mathbf{v}}_{\beta 1}) - \mu\mathbf{L}_{(1/\bar{\rho}_{\beta 1})}\bar{\mathbf{v}}_{\beta 1} + \mathbf{G}_{(1/\bar{\rho}_{\beta 1})}\bar{p}_{\beta 1} + \\ + \mathbf{G}_{(1/\bar{\rho}_{\beta 1})}\frac{\bar{p}_{\beta 1} - \bar{\rho}_{\beta 1} R\bar{T}_{\beta 1}}{R\bar{T}_{\beta 1}} \tag{4.141}$$

$$\bar{\mathbf{v}}_{\beta 2} = \bar{\mathbf{v}}_n + \mathbf{M}^{-1}\frac{\Delta t}{2}\mathbf{r}_2 \tag{4.142}$$

$$\mathbf{r}_3 = \mathbf{F} - \mathbf{K}(\bar{\mathbf{v}}_{\beta 2})(\bar{\mathbf{v}}_{\beta 2}) - \mu\mathbf{L}_{(1/\bar{\rho}_{\beta 2})}\bar{\mathbf{v}}_{\beta 2} + \mathbf{G}_{(1/\bar{\rho}_{\beta 2})}\bar{p}_{\beta 2} + \\ + \mathbf{G}_{(1/\bar{\rho}_{\beta 2})}\frac{\bar{p}_{\beta 2} - \bar{\rho}_{\beta 2} R\bar{T}_{\beta 2}}{R\bar{T}_{\beta 2}} \tag{4.143}$$

$$\bar{\mathbf{v}}_{\beta 3} = \bar{\mathbf{v}}_n + \mathbf{M}^{-1}\Delta t\mathbf{r}_3 \tag{4.144}$$

$$\mathbf{r}_4 = \mathbf{F} - \mathbf{K}(\bar{\mathbf{v}}_{\beta 3})(\bar{\mathbf{v}}_{\beta 3}) - \mu\mathbf{L}_{(1/\bar{\rho}_{\beta 3})}\bar{\mathbf{v}}_{\beta 3} + \mathbf{G}_{(1/\bar{\rho}_{\beta 3})}\bar{p}_{n+1} \\ + \mathbf{G}_{(1/\bar{\rho}_{\beta 3})}\frac{\bar{p}_{n+1} - \bar{\rho}_{n+1} R\bar{T}_{n+1}}{R\bar{T}_{n+1}} \tag{4.145}$$

$$\tag{4.146}$$

Next we assume a linear variation of pressure between t_n and t_{n+1} (similarly to the incompressible

[8] NB in the current work we have also tried the Crank-Nicolson scheme and actually for the examples solved, we could not detect any considerable differences.

[9] note that

- a) the term $\frac{\partial \mathbf{R}_m}{\partial \rho}\frac{\mathbf{R}_s}{RT} = \frac{\partial \mathbf{R}_m}{\partial p}\frac{p}{\rho}\frac{\mathbf{R}_s}{RT} = \frac{\partial \mathbf{R}_m}{\partial p}(p - \rho RT)$ contributes to the last term in the residuals
- b) the body force $\mathbf{F}$ (gravity) being constant has no time step index

case)[10] This assumption implies that $\bar{p}_{\beta 1} = p(t_n + \frac{\Delta t}{2}) = 0.5(p_n + \bar{p}_{n+1})$ and $\bar{p}_{\beta 2} = p(t_n + \frac{\Delta t}{2}) = 0.5(p_n + \bar{p}_{n+1})$.

This in turn allows us to write

$$\frac{1}{6}\left[\mathbf{G}_{(1/\bar{\rho}_n)}\bar{p}_n + 2\mathbf{G}_{(1/\bar{\rho}_{\beta 1})}\bar{p}_{\beta 1} + 2\mathbf{G}_{(1/\bar{\rho}_{\beta 2})}\bar{p}_{\beta 2} + \mathbf{G}_{(1/\bar{\rho}_{\beta 3})}\bar{p}_{\beta 3}\right] \approx \frac{1}{2}\left[\mathbf{G}_{(1/\bar{\rho}_n)}\bar{p}_n + \mathbf{G}_{(1/\bar{\rho}_{n+1})}\bar{p}_{n+1}\right] \tag{4.147}$$

Additionally, if we assume that the state equation residual is linear between t_n and t_{n+1}, then the fact that $\mathbf{R}_s(p_n, \bar{\rho}_n, \bar{T}_n) = 0$ and $\mathbf{R}_s(\bar{p}_{n+1}, \bar{\rho}_{n+1}, \bar{T}_{n+1}) = 0$ implies that $\mathbf{R}_s(t + \frac{\Delta t}{2}) = \bar{p}_{\beta 1} - \bar{\rho}_{\beta 1} R\bar{T}_{\beta 1} = 0$ and $\mathbf{R}_s(t + \frac{\Delta t}{2}) = \bar{p}_{\beta 2} - \bar{\rho}_{\beta 2} R\bar{T}_{\beta 2} = 0$. Therefore

$$\begin{aligned} \mathbf{G}_{(1/\bar{\rho}_n)}\frac{\bar{p}_n - \bar{\rho}_n R\bar{T}_n}{R\bar{T}_n} = \mathbf{G}_{(1/\bar{\rho}_{\beta 1})}\frac{\bar{p}_{\beta 1} - \bar{\rho}_{\beta 1} R\bar{T}_{\beta 1}}{R\bar{T}_{\beta 2}} = \\ = \mathbf{G}_{(1/\bar{\rho}_{\beta 2})}\frac{\bar{p}_{\beta 2} - \bar{\rho}_{\beta 2} R\bar{T}_{\beta 2}}{R\bar{T}_{\beta 2}} = \mathbf{G}_{(1/\bar{\rho}_{n+1})}\frac{\bar{p}_{n+1} - \bar{\rho}_{n+1} R\bar{T}_{n+1}}{R\bar{T}_{n+1}} = 0 \end{aligned} \tag{4.148}$$

Now we shall take out the terms multiplying the gradient of pressure outside the residuals (which is convenient in perspective of applying the fractional step technique), thus giving the final form of the time integrated momentum equation:

$$\begin{aligned} \bar{\mathbf{R}}_m = \mathbf{M}\frac{\bar{\mathbf{v}}_{n+1} - \bar{\mathbf{v}}_n}{\Delta t} - \frac{1}{6}[\hat{\mathbf{r}}_1 + 2\hat{\mathbf{r}}_2 + 2\hat{\mathbf{r}}_3 + \hat{\mathbf{r}}_4] - \\ - \frac{1}{2}\left(\mathbf{G}_{(1/\bar{\rho}_{n+1})}\bar{p}_{n+1} - \mathbf{G}_{(1/\bar{\rho}_n)}\bar{p}_n\right) \end{aligned} \tag{4.149}$$

with the modified residuals defined as

$$\hat{\mathbf{r}}_1 = \mathbf{F} - \mathbf{K}(\bar{\mathbf{v}}_n)(\bar{\mathbf{v}}_n) - \mu\mathbf{L}_{(1/\bar{\rho}_n)}\bar{\mathbf{v}}_n \tag{4.150}$$

$$\hat{\mathbf{r}}_2 = \mathbf{F} - \mathbf{K}(\bar{\mathbf{v}}_{\beta 1})(\bar{\mathbf{v}}_{\beta 1}) - \mu\mathbf{L}_{(1/\bar{\rho}_{\beta 1})}\bar{\mathbf{v}}_{\beta 1} \tag{4.151}$$

$$\hat{\mathbf{r}}_3 = \mathbf{F} - \mathbf{K}(\bar{\mathbf{v}}_{\beta 2})(\bar{\mathbf{v}}_{\beta 2}) - \mu\mathbf{L}_{(1/\bar{\rho}_{\beta 2})}\bar{\mathbf{v}}_{\beta 2} \tag{4.152}$$

$$\hat{\mathbf{r}}_4 = \mathbf{F} - \mathbf{K}(\bar{\mathbf{v}}_{\beta 3})(\bar{\mathbf{v}}_{\beta 3}) - \mu\mathbf{L}_{(1/\bar{\rho}_{\beta 3})}\bar{\mathbf{v}}_{\beta 3} \tag{4.153}$$

[10] We remind that this way we avoid the necessity for solving the continuity equation at every intermediate step of the Runge-Kutta scheme, an option not favorable due to associated computational effort increment.

Continuity equation Application of the Backward Euler scheme to the continuity Eq. (4.117) gives the following residual

$$\mathbf{R}_c = \frac{\rho_{n+1} - \rho_n}{\Delta t} + \rho_{n+1}\nabla \cdot \mathbf{v}_{n+1} + \mathbf{v}_{n+1} \cdot \nabla \rho_{n+1} + \frac{\partial \mathbf{R}_c}{\partial \rho} \frac{p_{n+1} - \rho_{n+1} R T_{n+1}}{R T_{n+1}} \tag{4.154}$$

The discrete version of the residual is

$$\bar{\mathbf{R}}_c = \mathbf{M}\frac{\bar{\rho}_{n+1} - \bar{\rho}_n}{\Delta t} + \mathbf{D}_{(\bar{\rho}_{n+1})}\bar{\mathbf{v}}_{n+1} + \mathbf{C}_{(\bar{\mathbf{v}}_{n+1})}\bar{\rho}_{n+1} + \frac{\partial \bar{\mathbf{R}}_c}{\partial \rho} \frac{\bar{p}_{n+1} - \bar{\rho}_{n+1} R \bar{T}_{n+1}}{R \bar{T}_{n+1}} \tag{4.155}$$

Since the state equation is satisfied exactly at t_{n+1}, the last term in equation (4.155) vanishes, giving

$$\bar{\mathbf{R}}_c = \mathbf{M}\frac{\bar{\rho}_{n+1} - \bar{\rho}_n}{\Delta t} + \mathbf{D}_{(\bar{\rho}_{n+1})}\bar{\mathbf{v}}_{n+1} + \mathbf{C}_{(\bar{\mathbf{v}}_{n+1})}\bar{\rho}_{n+1} \tag{4.156}$$

Next we evaluate the partial derivatives of the discrete residual in order to plug them in Eq. (4.130) (second line of the matrix equation):

$$\frac{\partial \bar{\mathbf{R}}_c}{\partial \bar{\mathbf{v}}} = \mathbf{D}_{(\bar{\rho}_n)} \tag{4.157}$$

$$\frac{1}{R\bar{T}_{n+1}} \frac{\partial \bar{\mathbf{R}}_c}{\partial \bar{\rho}} = \frac{1}{R}\left[\frac{1}{\Delta t}\mathbf{M}_{\left(1/\bar{T}_{n+1}\right)} + \mathbf{C}_{\left(\bar{\mathbf{v}}^i_{n+1}/\bar{T}_{n+1}\right)}\right] \tag{4.158}$$

$$\frac{\partial \bar{\mathbf{R}}_c}{\partial \bar{p}} = 0 \tag{4.159}$$

Now we introduce the residual (4.154) and the derivatives (4.157-4.159) into Eq. (4.130)(second line of the matrix equation) and obtain the the final form of the discrete continuity equation as:

$$\begin{pmatrix} \mathbf{D}_{(\bar{\rho}_n)} & \frac{1}{R}\left[\mathbf{M}_{1/\bar{T}_{n+1}} + \mathbf{C}_{\left(\bar{\mathbf{v}}_{n+1}/\bar{T}_{n+1}\right)}\right] \end{pmatrix} \begin{pmatrix} d\bar{\mathbf{v}} \\ d\bar{p} \end{pmatrix} = \tag{4.160}$$

$$= \mathbf{M}\frac{\bar{\rho}_{n+1} - \bar{\rho}_n}{\Delta t} + \mathbf{D}_{(\bar{\rho}_{n+1})}\bar{\mathbf{v}}_{n+1} + \mathbf{C}_{(\bar{\mathbf{v}}_{n+1})}\bar{\rho}_{n+1}$$

Fractional step split The discrete form of the coupled system Eq. (4.130) is the combination of Eqs. (4.149) and Eq. (4.160). In order to decouple the solution for the velocity and pressure, we apply the fractional step split. The fractional and end-of-step momentum equations can be thus written as:

$$\mathbf{M}\frac{\tilde{\mathbf{v}}-\bar{\mathbf{v}}_n}{\Delta t} = [\tilde{\mathbf{r}}_1 + 2\tilde{\mathbf{r}}_2 + 2\tilde{\mathbf{r}}_3 + \tilde{\mathbf{r}}_4] + \mathbf{G}_{(1/\bar{\rho}_n)}\bar{p}_n \tag{4.161}$$

$$\mathbf{M}\frac{\bar{\mathbf{v}}_{n+1}-\tilde{\mathbf{v}}}{\Delta t} + \frac{1}{2}\left(\mathbf{G}_{(1/\bar{\rho}_{n+1})}\bar{p}_{n+1} - \mathbf{G}_{(1/\bar{\rho}_n)}\bar{p}_n\right) = 0 \tag{4.162}$$

The residuals and corresponding intermediate fractional velocities are defined now as:

$$\tilde{\mathbf{r}}_1 = \mathbf{F} - \bar{\mathbf{K}}_{\bar{\mathbf{v}}_n}\bar{\mathbf{v}}_n - \mu\mathbf{L}_{(1/\bar{\rho}_n)}\bar{\mathbf{v}}_n \tag{4.163}$$

$$\tilde{\mathbf{v}}_{\beta 1} = \bar{\mathbf{v}}_n + \mathbf{M}^{-1}\frac{\Delta t}{2}\tilde{\mathbf{r}}_1 \tag{4.164}$$

$$\tilde{\mathbf{r}}_2 = \mathbf{F} - \bar{\mathbf{K}}_{\tilde{\mathbf{v}}_{\beta 1}}\tilde{\mathbf{v}}_{\beta 1} - \mu\mathbf{L}_{(1/\bar{\rho}_n)}\tilde{\mathbf{v}}_{\beta 1} \tag{4.165}$$

$$\tilde{\mathbf{v}}_{\beta 2} = \bar{\mathbf{v}}_n + \mathbf{M}^{-1}\frac{\Delta t}{2}\tilde{\mathbf{r}}_2 \tag{4.166}$$

$$\tilde{\mathbf{r}}_3 = \mathbf{F} - \bar{\mathbf{K}}_{\tilde{\mathbf{v}}_{\beta 2}}\tilde{\mathbf{v}}_{\beta 2} - \mu\mathbf{L}_{(1/\bar{\rho}_n)}\tilde{\mathbf{v}}_{\beta 2} \tag{4.167}$$

$$\tilde{\mathbf{v}}_{\beta 3} = \bar{\mathbf{v}}_n + \mathbf{M}^{-1}\Delta t\tilde{\mathbf{r}}_3 \tag{4.168}$$

$$\tilde{\mathbf{r}}_4 = \mathbf{F} - \bar{\mathbf{K}}_{\tilde{\mathbf{v}}_{\beta 3}}\tilde{\mathbf{v}}_{\beta 3} - \mu\mathbf{L}_{(1/\bar{\rho}_n)}\tilde{\mathbf{v}}_{\beta 3} \tag{4.169}$$

$$\tag{4.170}$$

Using (4.162) we express the end-of-step velocity as a function of the intermediate velocity:

$$\bar{\mathbf{v}}_{n+1} = \frac{\Delta t}{2}\mathbf{M}^{-1}\left[\mathbf{G}_{(1/\bar{\rho}_n)}\bar{p}_n - \mathbf{G}_{(1/\bar{\rho}_{n+1})}\bar{p}_{n+1}\right] + \tilde{\mathbf{v}} \tag{4.171}$$

and substitute it in the continuity equation (4.160) to obtain:

$$\begin{aligned}\mathbf{D}_{\bar{\rho}_n}\left[\tilde{\mathbf{v}} - \frac{\Delta t}{2}\mathbf{M}^{-1}\left(\mathbf{G}_{(1/\bar{\rho}_{n+1})}\bar{p}_{n+1} - \mathbf{G}_{(1/\bar{\rho}_n)}\bar{p}_n\right)\right] + \\ + \frac{1}{R\Delta t}\mathbf{M}_{(1/\bar{T}_{n+1})}\bar{p}_{n+1} - \frac{1}{\Delta t}\mathbf{M}\bar{\rho}_n + \mathbf{C}_{(\tilde{\mathbf{v}}/\bar{T}_{n+1})}\bar{p}_{n+1} = 0\end{aligned} \tag{4.172}$$

As usually we approximate the discrete Laplacian as $\mathbf{D}\mathbf{M}^{-1}\mathbf{G} \approx \mathbf{L}$. Additionally we assume that $\mathbf{L}_{(1/\bar{\rho}_{n+1})} \approx \mathbf{L}_{(1/\bar{\rho}_n)}$ and $\mathbf{D}_{\rho_{n+1}} \approx \mathbf{D}_{\rho_n}$

Thus, the final set of fractional equations for the compressible case reads

$$\mathbf{M}\frac{\tilde{\mathbf{v}}-\bar{\mathbf{v}}_n}{\Delta t} = \frac{1}{6}[\tilde{\mathbf{r}}_1(\bar{\mathbf{v}}_n) + 2\tilde{\mathbf{r}}_2(\bar{\mathbf{v}}_{\beta 1}) + 2\tilde{\mathbf{r}}_3(\bar{\mathbf{v}}_{\beta 2}) + \tilde{\mathbf{r}}_4(\bar{\mathbf{v}}_{\beta 3})] + \mathbf{G}_{(1/\bar{\rho}_n)}\bar{p}_n \tag{4.173}$$

$$\frac{1}{2}\mathbf{L}_{(\rho_n)}\left(\bar{p}_{n+1}-\bar{p}_n\right)+\frac{1}{R\Delta t}\mathbf{M}_{(1/\bar{T}_{n+1})}\bar{p}_{n+1}+\mathbf{C}_{(\tilde{\mathbf{v}}/\bar{T}_{n+1})}\bar{p}_{n+1})= \\ =\frac{1}{\Delta t}\mathbf{M}\bar{\rho}_n+\mathbf{D}_{\bar{\rho}_n}\tilde{\mathbf{v}} \tag{4.174}$$

$$\mathbf{M}\frac{\bar{\mathbf{v}}_{n+1}-\tilde{\mathbf{v}}}{\Delta t}+\frac{1}{2}\left(\mathbf{G}_{(1/\bar{\rho}_{n+1})}\bar{p}_{n+1}-\mathbf{G}_{(1/\bar{\rho}_n)}\bar{p}_n\right)=0 \tag{4.175}$$

Solution algorithm At this point we can summarize the obtained results in the algorithm of Table 4.2:

1. Solve the energy equation explicitly (Eq. (4.134)) and obtain the temperature $\bar{T}_{n+1} = f(\bar{\mathbf{v}}_n, \bar{\rho}_n, \bar{T}_n)$
2. Solve the fractional momentum equation (Eq. (4.173)), obtain $\tilde{\mathbf{v}}$ (using $\bar{\rho} = \bar{\rho}_n$ for all the intermediate steps)
3. Solve the continuity equation for the pressure (Eq. (4.174)), obtain $\bar{p}_{n+1}$
4. Obtain $\bar{\rho}_{n+1}$ from the ideal gas equation (Eq. (4.28))
5. Solve for end-of-step momentum (Eq. (4.175)) and obtain $\bar{\mathbf{v}}_{n+1}$

Table 4.2: Implementation of the compressible subsonic flow solver using an explicit solution of the energy equation

Remark: an iterative approach It is also possible to define an iterative approach following ideas very similar to the above. This is achieved by repeating the solution of the energy equation updating the density, velocity and pressure that are obtained as the solution of the momentum/continuity system. The discrete equations derived above hold, with the only difference that the $\bar{T}_{n+1}$ must be understood as a guess for the end-of-step temperature $\bar{T}^g_{n+1}$ that shall be updated. We summarize this approach in the algorithm, presented in Table 4.3:

4.5 Examples

In this section we shall test the functionality of the compressible sub-sonic solver in two benchmark examples. The first one deals with a square heated cavity and the second one with an elongated 8:1 cavity with the height being eight times larger than the width. The examples shall serve for validating both in the qualitative and quantitative senses. For example, some solvers fail to predict

1. Solve the energy equation (Eq. (4.134)), obtain the guess for the temperature, using a prediction for the density $\bar{\rho}_g$. $\bar{T}^0_{n+1} = f(\bar{\mathbf{v}}_n, \bar{\rho}_n, T_n)$
2. Begin the iterative loop till convergence in temperature
   a) Solve the fractional momentum equation (Eq. (4.173)), obtain $\tilde{\mathbf{v}}$
   b) Solve the compressible continuity equation for the pressure (Eq. (4.174)) obtain $\bar{p}^0_{n+1}$
   c) Obtain $\bar{\rho}^i_{n+1}$ from the ideal gas equation (Eq. (4.28))
   d) Solve the end-of-step momentum (Eq. (4.175)), obtain $\bar{\mathbf{v}}^i_{n+1}$
   e) Solve energy equation (Eq. (4.134)), using $\rho_g = \rho^i_{n+1}$, $\bar{T}^i_{n+1} = f(\bar{\mathbf{v}}^i_{n+1}, \bar{\rho}_g, \bar{T}_n)$
   f) Go to a)

Table 4.3: Implementation of the compressible subsonic flow solver using an iterative strategy

non-steady behavior of the flows, leading to stationary solutions. In this sense, the qualitative aspect of the solver is its ability to capture the non-steady phenomena. The results obtained by applying the compressible solver will be compared both with the results in the literature, obtained by using the Boussinesq approximation and a low Mach number hypothesis.

In the examples one additional physical approximation was introduced: the viscous dissipation function Ψ of the energy equation can be neglected for flows characterized by low viscosity and low speed. In the case of air this assumption is justified in the sub-sonic regime.

4.5.1 Natural convection in a square cavity

The natural convection in a cavity problem defines a classical benchmark example for the thermally coupled flows simulation algorithms. The problem is sketched in Fig. 4.1. The isothermal vertical walls are kept at constant but different temperatures. Upper and lower walls are considered adiabatic, that is a zero heat flux boundary condition is prescribed. Homogeneous Dirichlet boundary conditions are prescribed everywhere on the boundary for the velocity. The problem is completely defined by two dimensionless parameters, the Prandtl number Pr and the Rayleigh number Ra. These numbers are defined as

$$Pr = \frac{c_p \nu \rho}{k} \tag{4.176}$$

and

$$Ra = Pr \cdot Gr \tag{4.177}$$

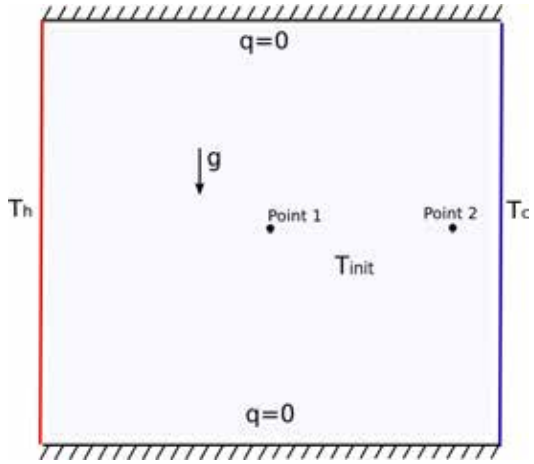


Figure 4.1: Model of a square heated cavity

with Gr being the Grashof number, defined as

$$Gr = \frac{gL^3(T_h - T_c)}{\nu^2(T_{av})} \tag{4.178}$$

where c_p is the specific isobaric heat, k is the conductivity, T_h and T_c are the temperatures of the hot and the cold walls respectively, L is the size of the cavity.

The first and simplest validation test consists in modeling this example for a relatively high Prandtl number ($\sim$ 1), (which is slightly higher than the physical value for the air ($\mathrm{Pr}_{air} = 0.7-0.8$)) the case where the stationary solution is obtained.

The left wall is heated isothermally to 20.5 C, and the right wall is heated isothermally to 19.5 C, the other sides being adiabatic. The fluid is initially at 20 C, which is the reference temperature. Ra and Pr are set to 10E6 and 1 respectively according to [74]. These settings correspond to a subcritical case where the stationary solution is obtained. Fig. 4.2 shows the initial temperature distribution and the temperature distribution at steady state, as well as steady state horizontal and vertical velocity profiles. The results exhibit excellent agreement with the ones presented in [75] (where the Boussinesq model was used in an Eulerian framework) and [74] (where this example was simulated using the Boussinesq-based PFEM model).

Heated cavity at low Prandtl number The heated cavity at high Prandtl number shown above defines the first step in validating the thermally coupled code. However, it is much more challenging to examine the settings where no steady-state is expected to be achieved. The numerical dissipation

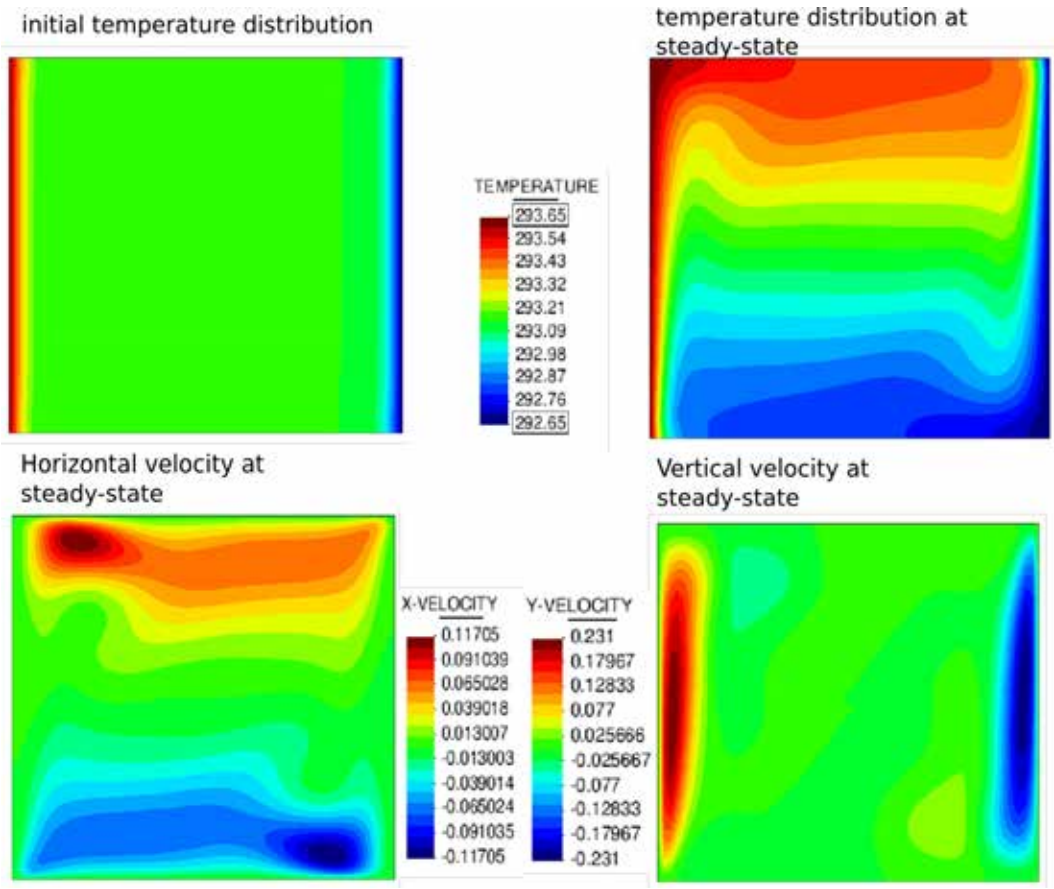


Figure 4.2: Square heated cavity for $Ra = 10E6$ and $Pr = 1$

Point	x	y
1	0.5	0.5
2	0.9	0.5

Table 4.4: Location of inspection points according to [4]

of the method, its accuracy and reliability can be tested in detail when a transient solution is analyzed.

To perform this step numerical results will be presented next for $Pr = 0.01$ and $Gr = 10E7$. Commonly modeled fluids, such as air or water have much higher values of Prandtl numbers, $Pr_{water} \approx 7.0$ and $Pr_{air} \approx 0.7$. For such Prandtl number, the fluid exhibits a steady-state solution that was shown above. However, it was discovered, that fluids with low Prandtl numbers (these are e.g. liquid semi-conductors), constitute an interesting problem: starting from $Gr \approx 5 \cdot 10E6$ the fluids start exhibiting an oscillatory flow field. The transient nature of the flow provides a much more complicated example for testing the solver. Computation of velocity and temperature fields as well as frequency spectra under such flow conditions shall give us a solid base for the validation of the solver developed in this work. In particular the presence of a periodic solution can give an insight of the quality of the model. As reported in literature, over-diffusive models lead to steady-state solutions.

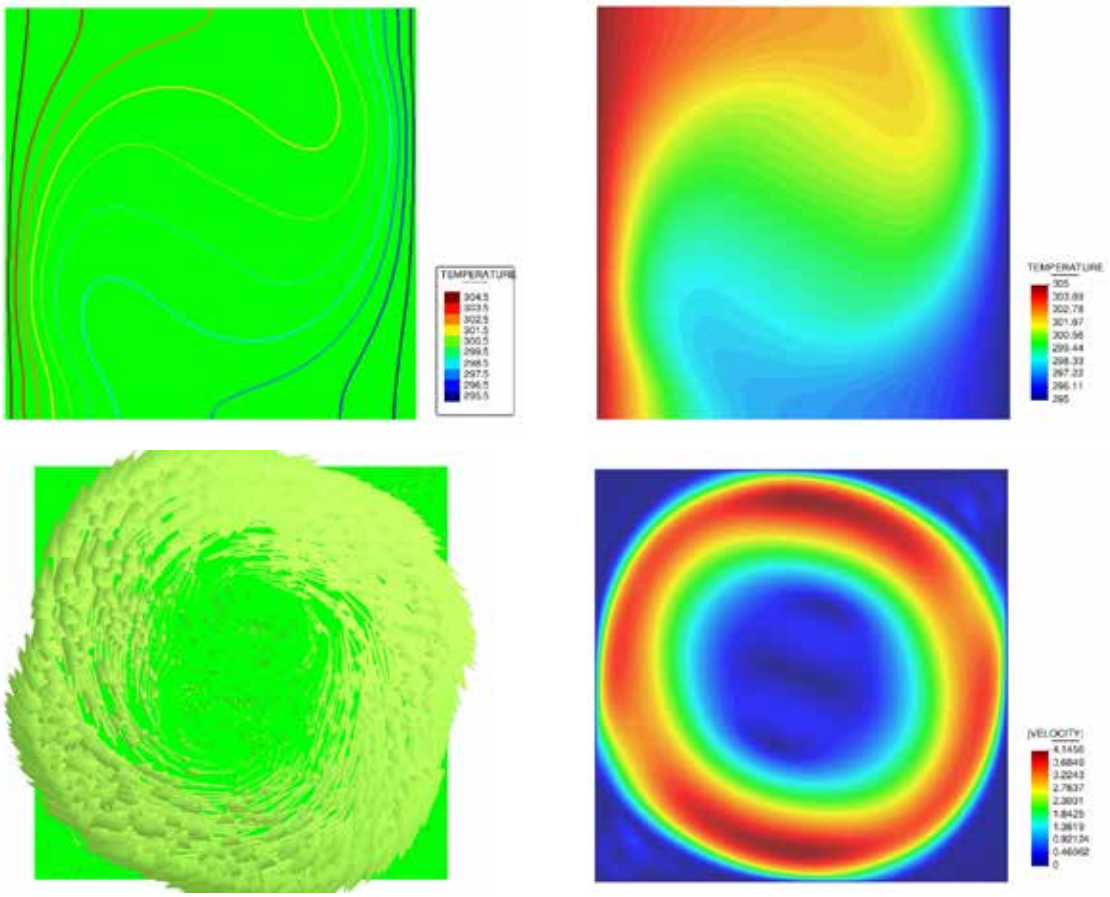
Remark It was observed that the problem is highly sensitive to the initial conditions. In particular solutions of the problem with initial temperature distribution set to the average T_{av} rather than the cold-wall temperature (T_c), show higher oscillations amplitude even on rough meshes.

The settings are chosen as follows:

- T_h=305 [K]
- T_c=295 [K]
- T_{init}=295 [K]
- Pr = 0.01
- Gr = 10E7

Fig. 4.3 illustrates the temperature and velocity distributions once a periodic behavior is reached. One can see a rotational velocity field.

As in the reference paper [4] two points of inspection are chosen (see Table 4.4):



(c) velocity vectors

(d) velocity distribution

Figure 4.3: Transient behavior in the heated cavity for Pr=0.01, Gr=10E7

Nondimensionalization was obtained by using the following scales: ΔT, $\frac{L^2}{\alpha}$, $\frac{\alpha}{L}$ for temperature, length, time and velocity respectively, where $\alpha = \frac{k}{c_p \rho}$ is the thermal diffusivity. Next we present figures displaying the horizontal velocity evolution at points 1 and 2 (see location of the inspection points in the Fig. 4.4).

Fig. 4.5 illustrates frequency spectra obtained on three different meshes (all meshes utilized were uniform unstructured meshes. Coarse mesh: 2500 elements, standard mesh: 5700 elements, fine mesh: 23000 elements). Comparing the frequency spectra with the results obtained by Mohamad and Viskanta, one can see that for the center of cavity the dominating frequency of ≈ 25 [Hz] is present. However, in our work we obtain one more frequency of ≈ 45 [Hz]. Similarly, at the second point (0.5, 0.9) the two dominating frequencies found in the numerical experiment of Mohamad and Viskanta [4] are found in the present work, however a third frequency of ≈ 70 [Hz] is also found.

Qualitative comparison in the time domain and quantitative comparison in frequency domain show good agreement with results of [4]. However there the time span of the analysis is not sufficiently large in order to gain a detailed insight regarding the periodic solution. In order to perform a more detailed comparison, we refer to [5], where detailed time histories of the periodic solution are given for this same example. Prandtl and Grashof numbers used in [5] are chosen as $Pr = 0.005$, $Gr = 5 \cdot E05$.

Detailed comparison Details with regard to periodic behavior (period, amplitude etc.) were presented for natural convection example in [5]. There the case with

- T_h=305 [K]
- T_c=295 [K]
- Gr=10E5
- Pr=0.005

was analyzed. Nondimensionalization of time and velocity are obtained by scaling with $\frac{H^2}{\alpha}$ and $\frac{\alpha}{H}$ (α and H stand for thermal diffusivity and cavity height) respectively. Temperature is scaled with $\Delta T = 10[K]$. Here we present the data for two different uniform unstructured meshes, one containing 36000 linear triangles (fine mesh) and the other one 15 000 linear triangles (standard mesh). The data was taken at the point (x,y)=(0.773,0.773) according to [5]. Fig. 4.6 displays the horizontal and vertical velocity as well as the temperature evolution once the periodic behavior is reached. The time window of 0.5 (nondimesnional time) is taken in both cases (from 8 to 8.5 in our work and from 5 to 5.5 in [5]. This is motivated by the fact that the periodic behavior is reached at different times. All the displayed quantities are nondimensional.

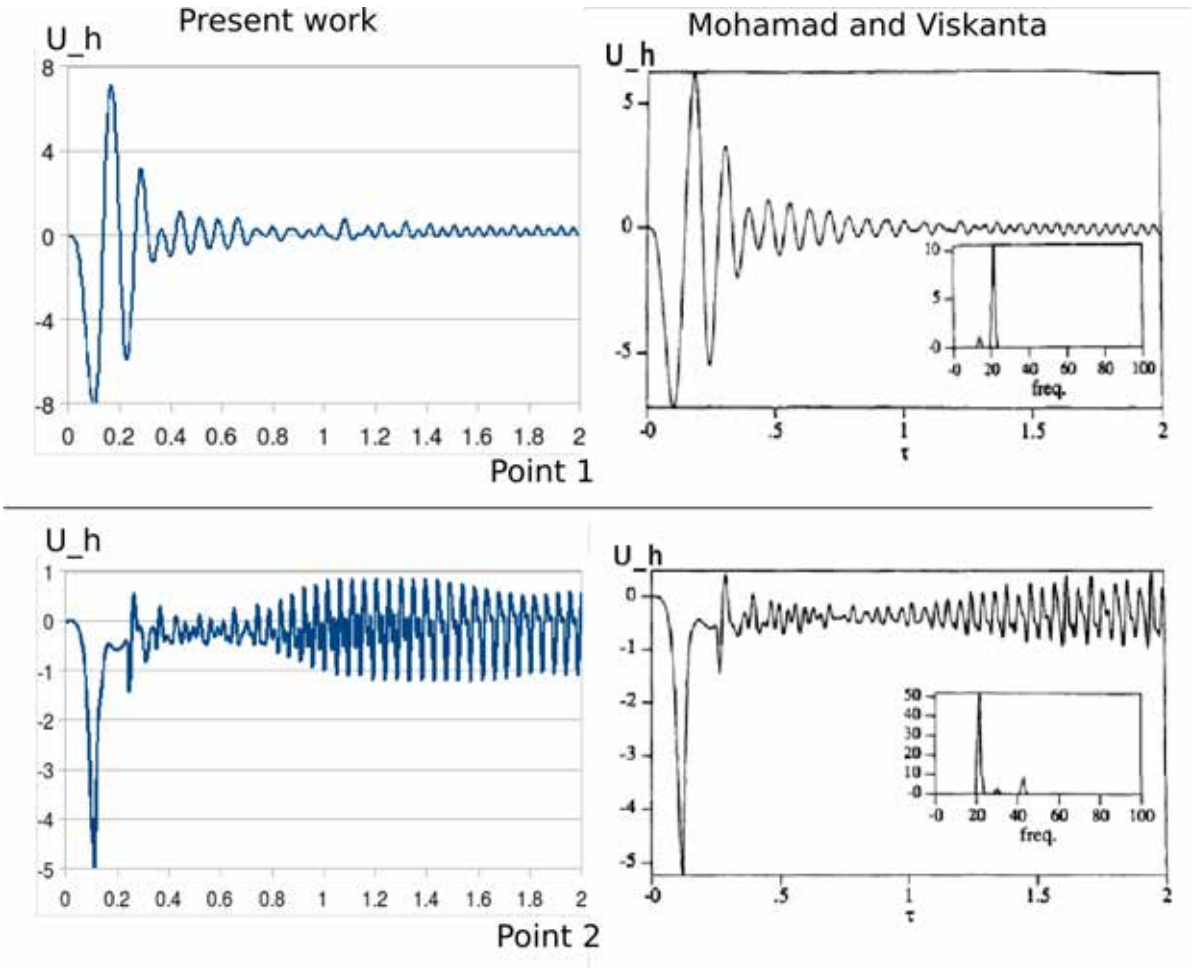


Figure 4.4: Velocity evolution and frequency spectra in the heated cavity for Pr=0.01, Gr=10E07. Comparison with [4]

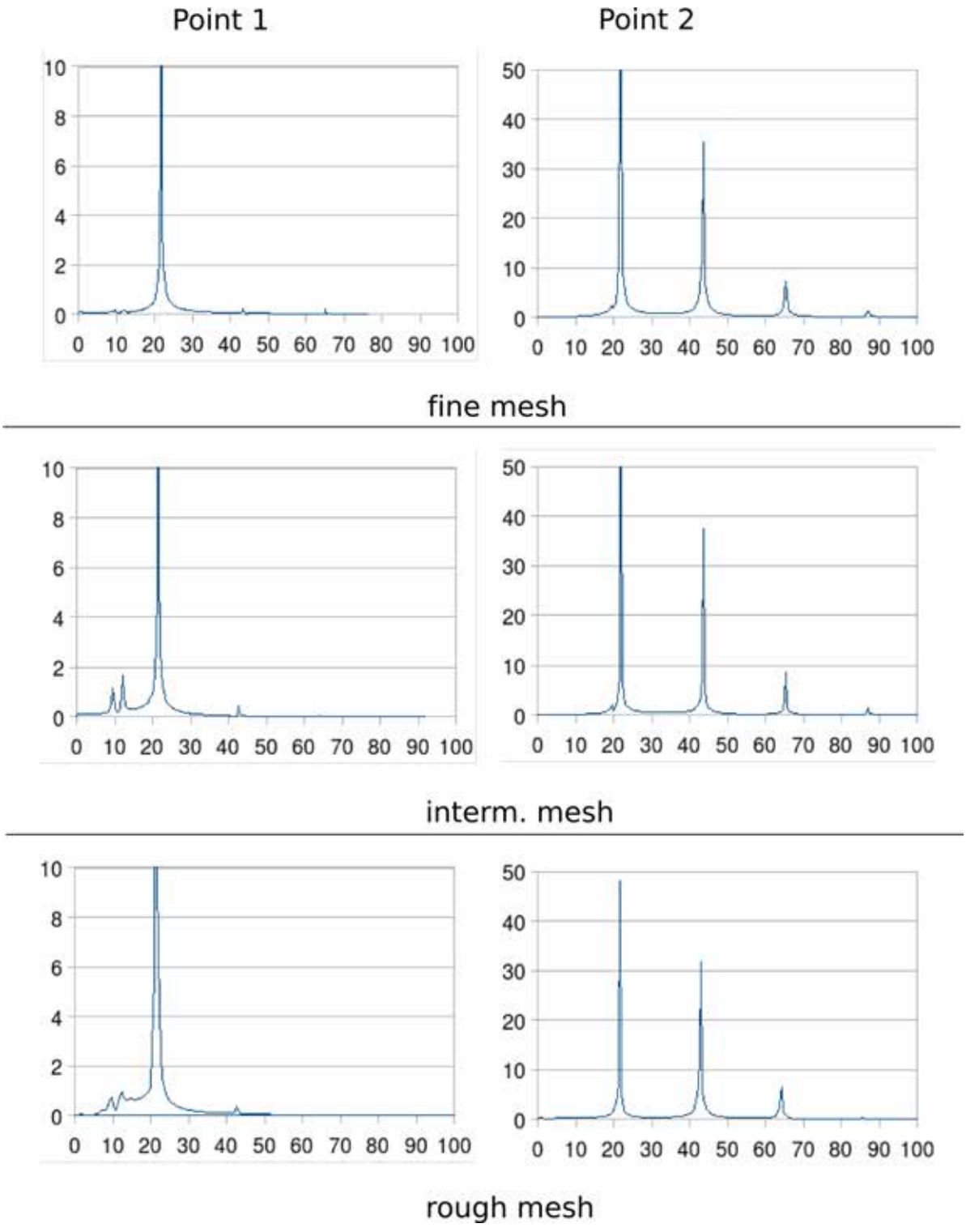


Figure 4.5: Frequency spectra in the heated cavity at inspection points (defined in Table 4.4) for different meshes

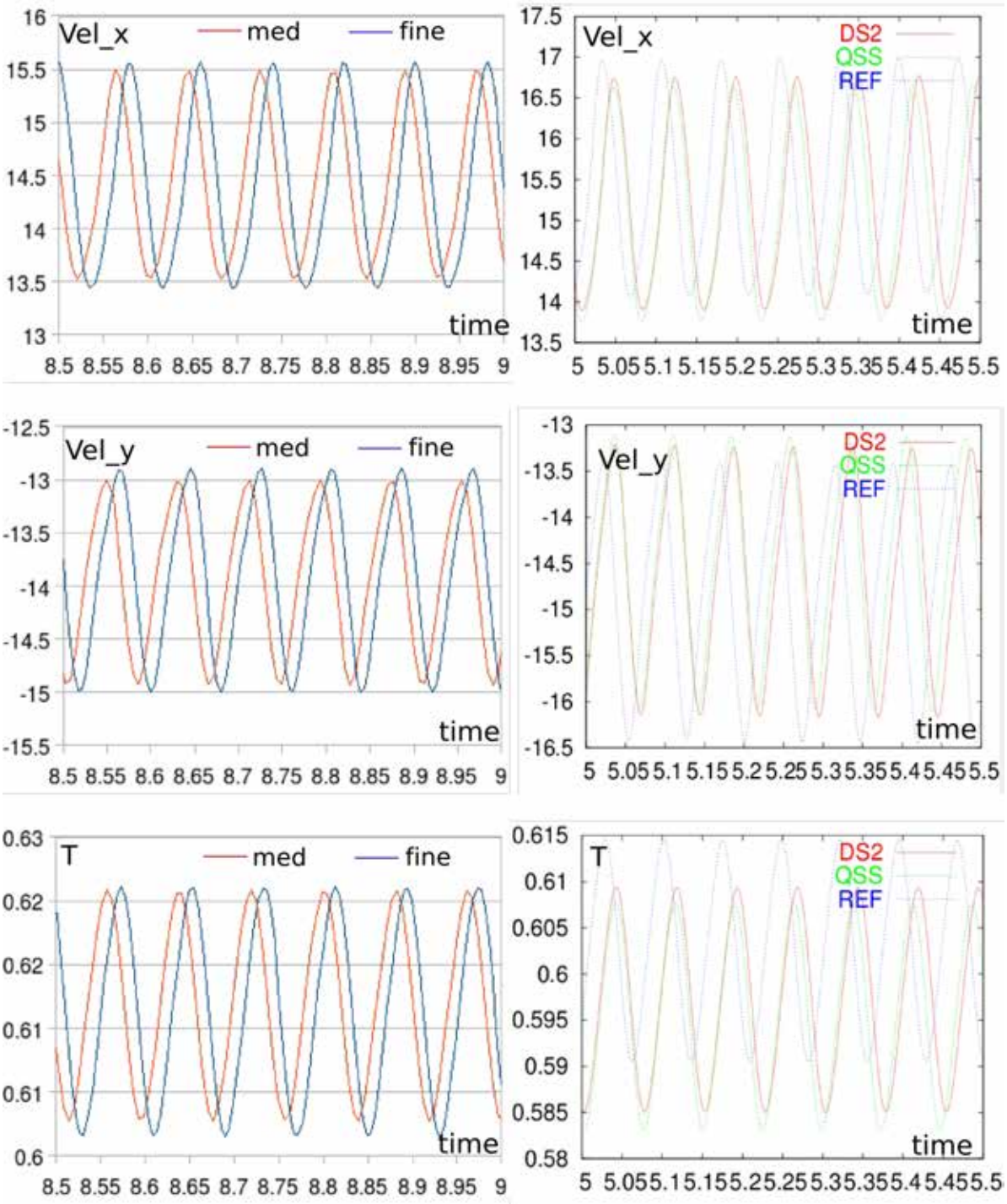


Figure 4.6: Natural cavity at Gr=10E5, Pr=0.005: results at (x,y)=(0.773,0773) (results in terms of nondimensionalized quantities)

parameter	present work (Compressible, GLS)	Principe (Boussinesq, OSS) [5]
Period avg	0.053	0.053
T avg	0.612	0.6025
T ampl	0.2	0.22
u_x avg	14.5	15.5
u_x ampl	2.4	2.8
u_y avg	-13.9	-14.8
u_y ampl	2.25	2.8

Table 4.5: Comparison of the compressible solver results with the ones obtained in [5]

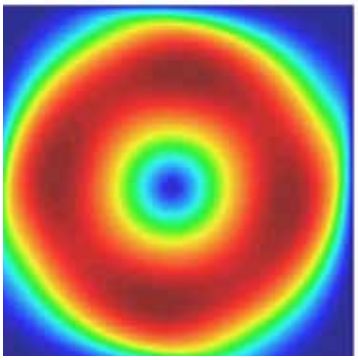
In the Table 4.5 we summarize the comparison with respect to nondimensionalized period, temperature, horizontal and vertical velocities.

Comparison with the results of Principe [5] shows an excellent agreement with respect to the period of oscillations. The average values of the temperature obtained here are slightly higher than in [5], whereas velocities are slightly lower. With respect to the average values, the discrepancy lies within the $\approx 5\%$ range. The amplitudes differ more, especially the amplitude of the vertical velocity, where the difference reaches $\approx 20\%$. Despite of the slight under-prediction of the amplitudes, that may be explained by additional diffusion of GLS stabilization in contrast with OSS as well as difference of the utilized elements (linear triangles here and bilinear quads in [5]), the results are in good agreement.

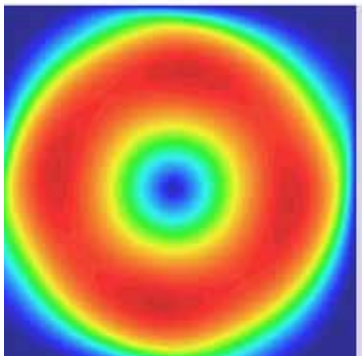
Comparison with a "low Mach model" All the examples discussed until now have shown the comparisons between the compressible solver derived here with results obtained by Boussinesq models in the literature. Next we show a comparison with the Low Mach number model, presented in [5]. The same heated unit cavity example is considered for simplicity at a low Rayleigh number $Ra = 10E3$. This shall facilitate approaching the steady-state quickly. The Prandtl number used in this work was set to $Pr = 0.01$, even though in [5] no value for it is provided making the quantitative comparison impossible. The interesting feature of this example is that the central vortex shifts its position as a function of $\epsilon = \frac{T_h - T_c}{T_h + T_c}$, which reflects the temperature difference. Results of the vortex position obtained using Low Mach number model were presented in [5]. It was noticed that Boussinesq solvers cannot capture this phenomenon and always result in a vortex positioned exactly in the center of cavity. This happens since the flow expansion due to heating is neglected when Boussinesq hypothesis is used. This can be seen in the Fig. 4.7.

To test our compressible solver four sets of temperature and the corresponding ϵ values were analyzed:

1. $T_h = 301$ [K], $T_c = 299$ [K], leading to $\epsilon \approx 0$



(a) Central vortex position for $\epsilon = 0.1$



(b) Central vortex position for $\epsilon = 0.5$

Figure 4.7: Position of the central vortex for $Pr = 0.01$ and $Ra = 3 \cdot E03$: Boussinesq model

2. $T_h = 330$ [K], $T_c = 270$ [K], leading to $\epsilon = 0.1$

3. $T_h = 750$ [K], $T_c = 250$ [K], leading to $\epsilon = 0.5$

4. $T_h = 1250$ [K], $T_c = 250$ [K], leading to $\epsilon = 0.66$

Next we present the velocity contours showing the position of the vortex in terms of ϵ for the compressible solver (velocity: min=0, max=6.0 [m/s]): Fig. 4.8.

The horizontal coordinates (x) of the central vortex are summarized next:

1. $\epsilon \approx 0$: x=0.5

2. $\epsilon = 0.1$: x=0.486

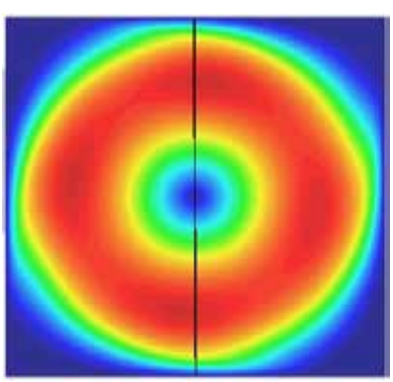
3. $\epsilon = 0.5$: x=0.4335

4. $\epsilon = 0.66$: x=0.41619

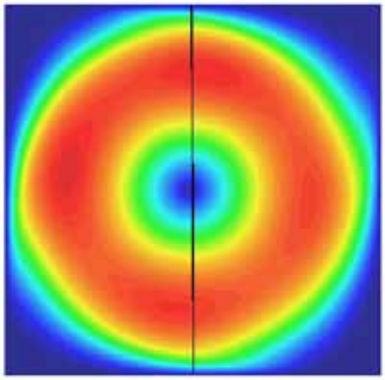
This example shows the possibility of using the fully compressible solver as an alternative to the low Mach number solver and its clear superiority in comparison with Boussinesq-based solvers.

4.5.2 8:1 heated cavity

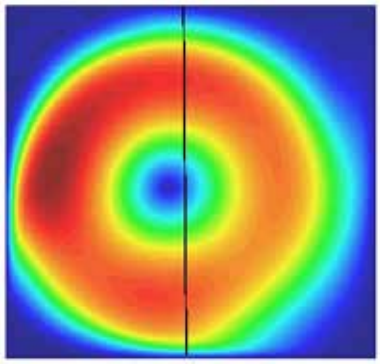
Next we present a benchmark proposed by Christon et al in [76]. The example is similar to the previous one, however the geometry of the cavity is different. The ratio between the vertical/horizontal edge length is 8:1. The spatial structure of the flow is made of vertical and horizontal boundary layers and corner structures, which depend very sensitively on the aspect ratio, Prandtl number and thermal boundary conditions. All these features lead to a very complex time behavior resulting



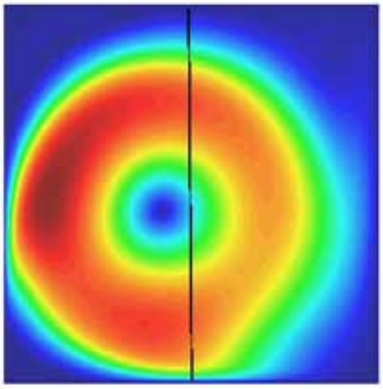
(a) Central vortex position for $\epsilon \approx 0$



(b) Central vortex position for $\epsilon = 0.1$

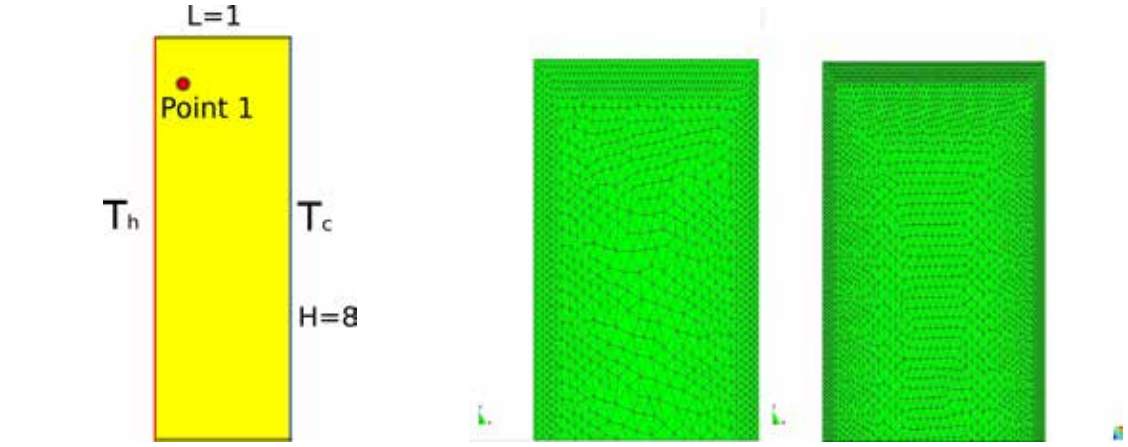


(c) Central vortex position for $\epsilon = 0.5$



(d) Central vortex position for $\epsilon = 0.66$

Figure 4.8: Compressible solver: position of the central vortex for $Pr = 0.01$ and $Ra = 3 \cdot E03$ as a function of ϵ.



(a) model and location of the reference point (b) meshes used

Figure 4.9: 8:1 differential heated cavity

from several instability mechanisms, travelling waves in the vertical boundary layers and thermal instabilities along the horizontal walls in particular, which can interact strongly with internal wave dynamics.

The geometry is shown in Fig. 4.9 and the benchmark parameters are summarized next:

- L = 1m
- H = 8 m
- $T_h = 285.15$ [K]
- $T_c = 275.15$ [K]
- $Ra = 3.4E05$
- $Pr = 0.71$
- Point of analysis: $(x, y) = (0.181, 7.37)$
- top and bottom walls are adiabatic.

The model is shown schematically in Fig. 4.9. Nondimensionalization was obtained by using the following scales: ΔT, L, $\sqrt{g\beta L\Delta T}$, $\frac{L}{\sqrt{g\beta L\Delta T}}$ for temperature, length, time and velocity respectively,

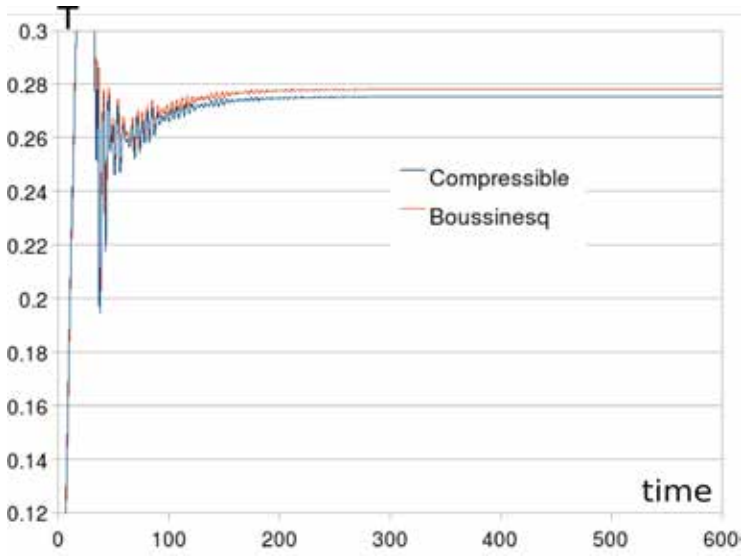


Figure 4.10: 8:1 cavity: Temperature (nondimensionalized) evolution for Ra=2.8 E05 (undercritical regime)

where $\beta = \frac{1}{T_{av}}$ is the thermal expansion coefficient. All the data will be presented in terms of nondimensionalized quantities. Tests were carried out on two meshes, one containing 17000 (referred as coarse) triangles with size varying from 0.02 (at the walls) to 0.08 (in the interiors) and the other one containing 45000 (referred as "fine") with sizes varying from 0.01 (at the walls) to 0.03 (in the interiors). Both meshes are shown in Fig. 4.9.

In establishing this benchmark Christon et al [76] propose $Ra = 3.4E05$ as it constitutes a slightly over-critical Rayleigh number. All the investigations carried out [6], [77], [78], [79] announce critical Rayleigh number to be above $3.0E05$. To ensure that in the undercritical regime the flow converges to the steady behavior, we performed a series of tests with $Ra = 2.8E05$. Both the fully compressible and Boussinesq solvers converged to a steady-state solution independent of the mesh used (this can be seen in Fig. 4.10, where the temperature evolution at "Point 1" defined by Christon [6]).

First of all the results (temperature evolution at point 1) of the Boussinesq-based solver developed here (semi-explicit, GLS-stabilized) are compared with those obtained using the Boussinesq-based implicit solver (using second order backward difference (BDF2) time integration scheme, OSS-stabilized). These are presented in the Fig. 4.11. On the rough mesh both methods fail to capture periodic solution, while the transitory region coincides for both methods. On a fine mesh the periodic solution is captured and the amplitudes are very similar.

Next we present the results obtained on two meshes with the semi-explicit compressible solver

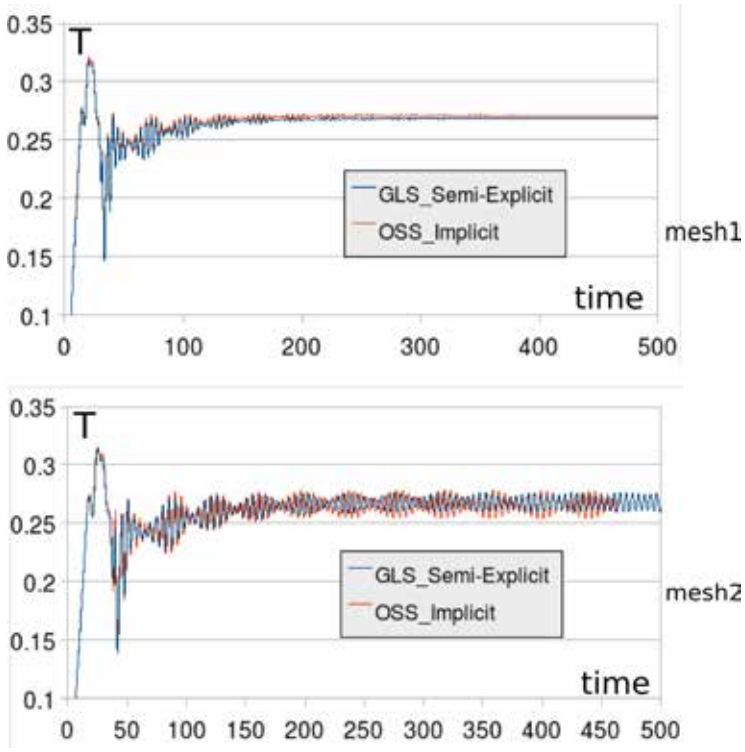


Figure 4.11: 8:1 cavity: Temperature (nondimensionalized) evolution for Ra=3.4 E05. Results obtained with two Boussinesq solvers: semi-explicit and implicit

parameter	Comp. Solver	Christon(ref.mesh)
Period avg	3.51	3.52
u_x avg	0.0608	0.06112
u_x ampl	0.0521	0.05078
u_y avg	0.4473	0.4638
u_y ampl	0.07724	0.0726
T avg	0.26588	0.2663
T ampl	0.0421	0.03975

Table 4.6: Comparison of the compressible solver results with the ones obtained in [6]

developed in this work. These are compared with the Boussinesq solver results and results found in the benchmark paper [76]. Fig. 4.12 shows the evolution of temperature at "Point 1" obtained on coarse and fine meshes using the semi-explicit Boussinesq and fully compressible solvers. The compressible solver was able to produce good results already on the coarse mesh. The comparison of the compressible solver results on coarse and fine meshes is shown in Fig. 4.13 (a). One can see that the oscillation amplitude does not vary between both meshes (see the superimposed graphs in Fig. 4.13). The average temperature value is found to be $\Theta = 0.265888$ and the average amplitude $a = 0.007377$. The period of oscillations is found to be 3.51. The comparison with the results of Christon et al in [76] are summarized in the Table 4.6.

The results provide some confidence in advantages of using the compressible solver.

It was observed in [6] that explicit algorithms on coarse meshes are unable to capture the oscillatory solution found at $Ra = 3.4 \cdot 10E05$ and $\mathrm{Pr} = 0.71$. The cause was attributed to the mass lumping used in the explicit schemes. However in the present work both fully implicit and semi-explicit Boussinesq solvers gave similar results, leading to a steady-state solution on coarse meshes and capturing the periodic behavior on finer meshes. We note that the compressible semi-explicit solver captured the oscillatory behavior already on the rough mesh. The mass lumping in the momentum equation was used also in the compressible solver, however the periodic solution was captured. The possibility of attributing the quality of the solution to the artificial diffusion introduced by stabilization also does not seem possible, as two types of Boussinesq solvers, one using the GLS method and the other using OSS method both failed to capture periodic solutions on a coarse mesh.

The stream-lines pattern obtained by solving the problem with the compressible solver is shown in Fig. 4.13 (b). The results match very well the results reported by Salinger et al. in [79], namely the stream-line pattern corresponding to the flow after the first Hopf bifurcation. [11]

[11] Hopf or Andronov-Hopf bifurcation is a local bifurcation in which a fixed point of a dynamical system loses stability as a pair of complex conjugate eigenvalues of the linearization around the fixed point cross the imaginary axis of the complex plane.

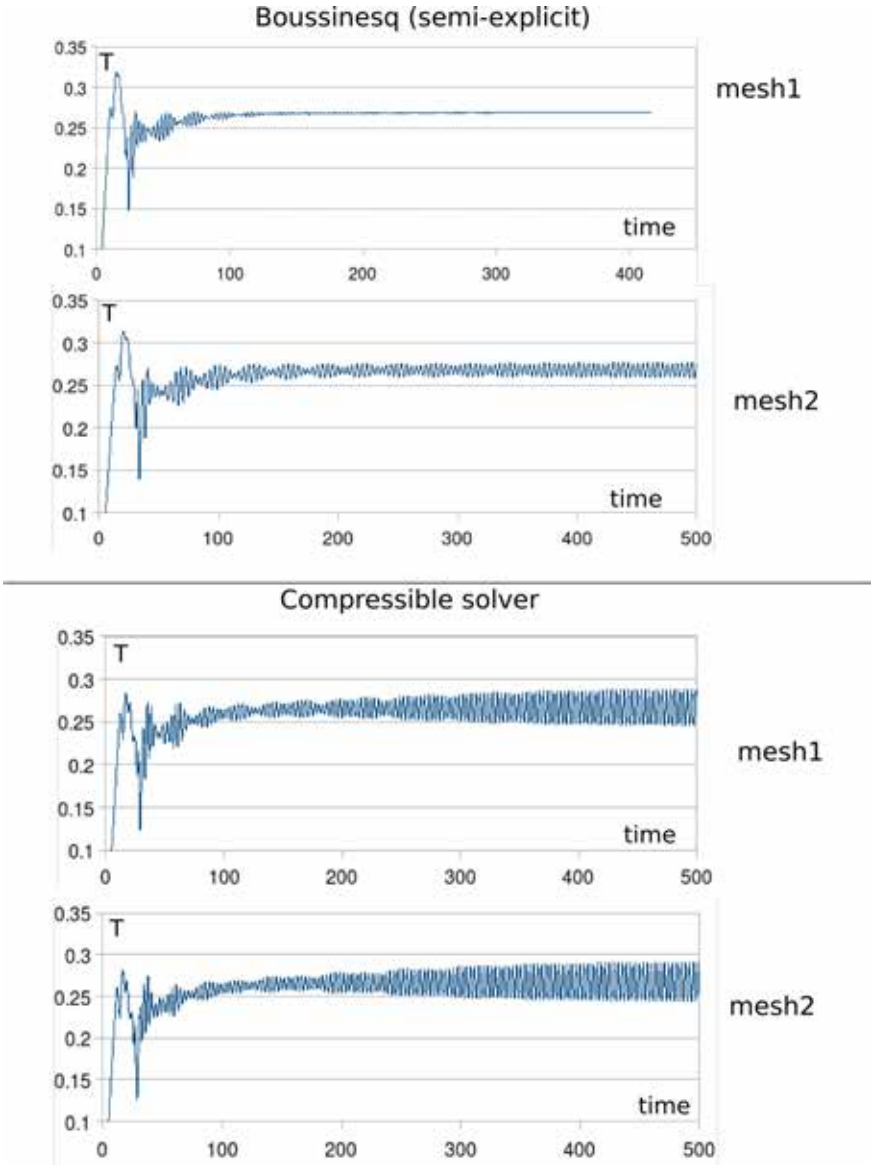
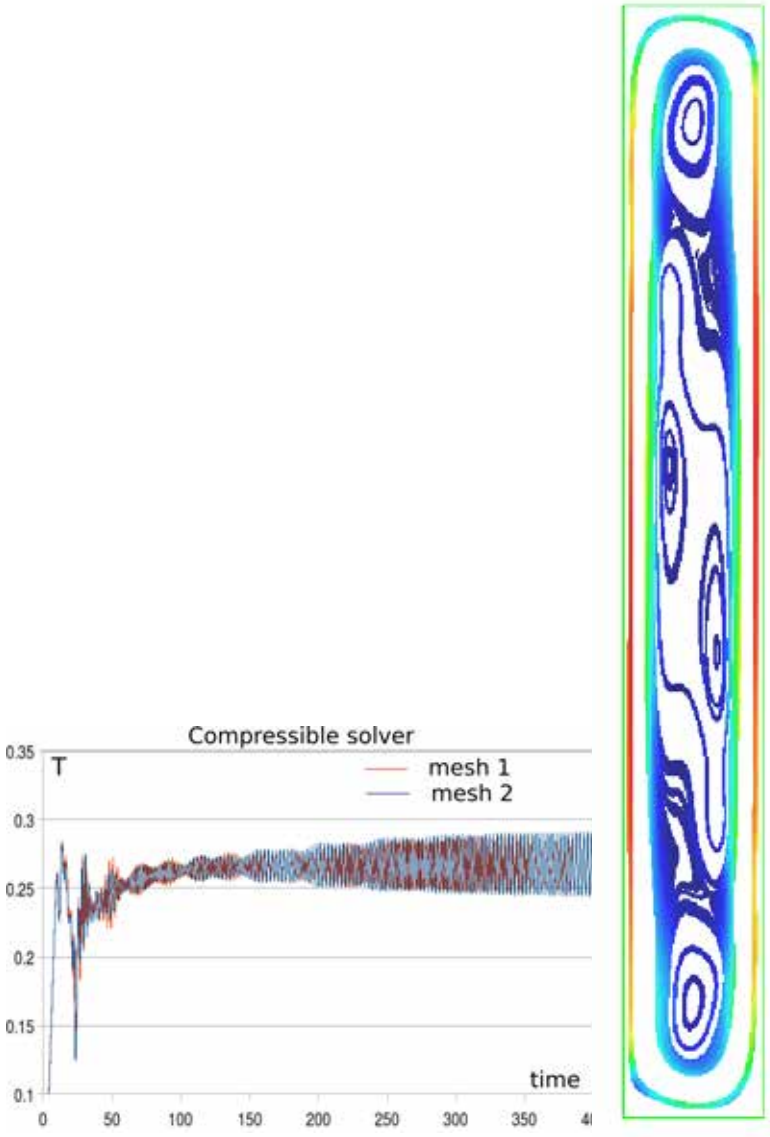


Figure 4.12: 8:1 cavity: Temperature (nondimensionalized) evolution for Ra=3.4 E05



(a) Results obtained on rough and fine meshes superimposed

(b) stream lines pattern

Figure 4.13: Compressible solver results. 8:1 differential heated cavity at Ra=3.4 E05.

The critical Rayleigh number announced in [79] is found to be $Ra = 3.61 \cdot 10E5$. However Xin and Le Quere [78] mentioned the critical Rayleigh number of $Ra = 3.1 \cdot 10E5$. Also in [76] the regime at $Ra = 3.1 \cdot 10E5$ is referred to as "slightly over-critical". In fact, all the results collected there prove that this regime is over-critical.

4.6 Summary and conclusions

In this chapter a finite element formulation and an algorithm for solution of compressible sub-sonic flows has been proposed. The formulation based on the complete set of compressible Navier-Stokes equations. Absence of the shock waves in the sub-sonic regime permitted the utilization of the non-conservation form of the governing equations (momentum and energy). This facilitated the implementation, as well as permitted the re-utilization of parts of the incompressible code. An efficient algorithm for the solution of the governing system of equations was introduced. We have shown that in case of explicit treatment of the energy equation, this becomes uncoupled from the rest of the system naturally (without the need for introducing any approximation).

For solution of (uncoupled) momentum/continuity system we proposed a semi-explicit method, where the momentum equation was integrated explicitly (except for the pressure term) and the continuity is integrated implicitly using a dissipative time integrator. For assuring accuracy we have chosen a 4th order Runge-Kutta scheme, omitting, for the sake of computational efficiency, the resolution of the continuity equation for the intermediate velocities.

The resulting method has the following features

- efficiency close to that of a fully explicit approach
- implicit treatment of pressure leading to the invariance of the time step estimation of the acoustic waves
- the frequency spectrum contains the acoustic contributions that can be correctly resolved by FEM
- spurious high frequencies are damped by a dissipative time integrator

We also addressed a possibility of an iterative approach, where the solution of the energy equation based on the known values of velocity, pressure, density is considered to be a guess for the momentum/equation system, that is updated until convergence in temperature is achieved.

In either case, the fractional step split was applied to the momentum/continuity system. The introduced methodology not only permits splitting the monolithic system into a set of smaller ones, but also a modular implementation similar to that of the incompressible case.

The algorithm was validated in several benchmarks examples, comparing the results with both Boussinesq-based and low Mach number-based solvers available in literature. The expansion of the flow, neglected when the Boussinesq hypothesis is implied, can be well predicted by the developed compressible model providing some confidence that it can be a competitive alternative to the low Mach number model for sub-sonic flows.

Chapter 5

Eulerian-Lagrangian thermo-mechanical coupling strategy

5.1 Introduction and outline

So far we have considered two different coupling strategies for multi-field problems: partitioned and monolithic Lagrangian-Lagrangian approaches. In this chapter we will derive a different approach, where the PFEM will be coupled to an Eulerian solver. This chapter leads to an efficient algorithm for the solution of an important practical problem which motivated this work, namely the simulation of polymer melting under fire conditions in living areas, which we will briefly describe next.

5.1.1 Motivation

Thermoplastic polymers are widely used in upholstered furniture, mattresses, and solid plastic assemblies. The necessity of predicting their behavior under fire conditions motivated the developments presented in this chapter. The distinguishing feature of heated/ignited thermoplastic polymers is their tendency to liquidity and flow, while practically not changing their density [80]. This leads to dripping, and further on to the production of liquid pools that can serve as the sources of further ignition. Therefore, the prediction of the melt spread is of high importance for determination of the danger zones in living areas [81]. The snap-shots depicting polymer-made domestic objects under

fire conditions are shown in Fig. 5.1.

(a) Burning mattress

(b) Burning plastic chair

Figure 5.1: Thermoplastic domestic objects under fire conditions

There exist by now no established methodology for simulation of this complex practical problem. First steps were made in [82] where polymer melting was modeled using the PFEMunder the assumption of the constant heat flux applied to the polymer surface. Ambient air was not taken into account. The results obtained in the numerical simulation were compared to a few experimental results available proving applicability of the PFEM in this field.

Contemporary to the developments of the present work, Marti et al. [83] developed an extension of [82], by adopting the ideas ofa unified formulation [21]. There, both ambient air and the polymer are solved monolithically using a Lagrangian PFEM approach, based on constant-pressure elements. The formulation was enriched with the Boussinesq modification and a simple combustion model.

Even though clearly defining an important step, the formulation in [83] has a number of drawbacks. First of all, the constant pressure formulation leads to under-estimation of velocity field for moderate to high values of the bulk modulus. This has a particularly strong impact upon the correct representation of air exposed to strong heating, that exhibits vortices and undergoes strong convection. On the contrary, utilization of low bulk modulus values can lead to excessive compression and thus volume loss (also crucial for the air as it has a low density in comparison with polymer).Additionally, the temperature increments expected in the problem of interest ($\Delta T \propto 300$ [K])make the utilization of Boussinesq assumption debatable.

We propose here an alternative that aims at minimizing the computational cost achieved by using the immersed boundary method approach, combining the Lagrangian formulation for the polymer and Eulerian one for the air. A fully compressible formulation for the air ensures its validity in the presence of high temperature alterations. The techniques and formulations developed in preceding chapters will be treated here as "ready-to-use" building blocks. We will put the emphasis on the

coupling between them. We must note that a combustion model is not included. Rather we present a general framework of thermo-mechanical coupling of the PFEM and an Eulerian formulation.

The chapter is organized as follows: first we motivate the developments of the chapter, highlighting difficulties of the modeling of the problem of interest and justifying the feasibility of the kinematic description choice (combination of a Lagrangian description for the polymer with the Eulerian description for the surrounding air).

Next we will specify the "building blocks" of the overall solution strategy, namely the chosen solvers and the couplings between them. We specify a partitioned approach, where polymer is modeled using quasi-incompressible PFEM (model developed in **Chapter 3**) and the ambient air is modeled using the compressible Eulerian solver (model developed in **Chapter 4**).

Then a mechanical coupling strategy, which adopts the philosophy of immersed boundary methods, will be derived. We shall introduce several simplifications on the basis of the physical properties of the constituents (polymer and air), justifying a one-way Dirichlet coupling. The imposition of the interface Dirichlet conditions in the weak sense is introduced next.

Afterwards a methodology for the solution of the thermal part of the problem (energy equation) will be introduced. We will resolve the energy equation for both the sub-domains (polymer and air) on the Eulerian mesh. Projection of the thermal properties between Eulerian and Lagrangian meshes that is necessary in such settings, will be presented.

Finally the overall solution algorithm will be presented.

The chapter concludes with validation tests of the coupling and a practical example, showing the potentiality of the method.

5.2 Model constituents

The simulation of the polymer melt flow problem is quite challenging as it must include models for for both the sub-systems (ambient air and the polymer) and the interaction between the sub-systems at both mechanical and thermal levels. The simulation of the entire problem can thus be defined as a solution of the following set of sub-problems

- Solution of thermally coupled flow problem for the polymer
- Solution of the thermally coupled flow for the ambient air
- Simulation of the thermal interaction between air and polymer
- Simulation of the mechanical interaction between air and polymer

Next we shall address the choice for modeling each one of these problems.

5.2.1 A mechanical solver for the polymer

The thermoplastic polymer melting simulation is rather complex by itself even if no interaction with the surrounding air is considered. The thermo-mechanical problem involves free surface flow, and this is additionally complicated by the material degradation [82],[81]. The important feature for modeling thermoplastics is that they can be viewed as frozen liquids, that behave as incompressible viscous fluids, with viscosity being a function of temperature. Indeed, viscosity is a key determinant of the melting behavior of thermoplastics. Temperature elevation both increases the mobility of the polymer chains and breaks chemical bonds to create smaller fragments; thus viscosity is a function of both molecular weight and temperature [84]. To avoid calculation of the molecular weight distribution of the polymer melt, which would greatly increase the complexity of the model, the molecular weight dependence is neglected and viscosity is assumed to depend exclusively upon the temperature. The viscosity-temperature curves determined empirically are available for polypropylene (one of the most commonly used thermoplastic) in [80]. Fig. 5.2 shows the temperature dependence of viscosity of polypropylene. The three curves reflect the polymer viscosity changes over the temperature range between 200 and 450 [C] for the undegraded polymer and the polymer obtained after solidification of the melt obtained by exposition to different heat fluxes. One can see that the irreversible change of the chemical structure effects the viscosity of the polymer especially at initial low temperatures.

From the point of view of the numerical solution, a complication arises as the viscosity assumed in parts of the polymer ranges over the span up to 6 orders of magnitude, for the temperature increments encountered in the problem of interest. On one hand this has an impact upon the system conditioning (the system matrix entries become "unbalanced"), thus affecting the convergence behavior of iterative solvers. On the other hand, these severe viscosity changes lead to a chaotic and discontinuous free surface evolution (creation of droplets, their separation together with the motion of the viscous bulk zone). This is schematically depicted in Fig. 5.3. A polymer bulk of rectangular shape fixed by a rigid holder on the right and bottom edges is exposed to constant heat flux.

Fig. 5.4 shows the viscosity distribution obtained in the simulation performed using PFEM in [80]. There only polymer melting was simulated (no interaction with the surrounding), and the constant heat flux applied to the free surface was assumed, the melt is captured in the heated catch-pan. We shall not discuss the quantitative results, the only aim is to show a typical viscosity distribution. One can see that the viscosity at the surface is of order 10E-04 [Pa s], whereas in the

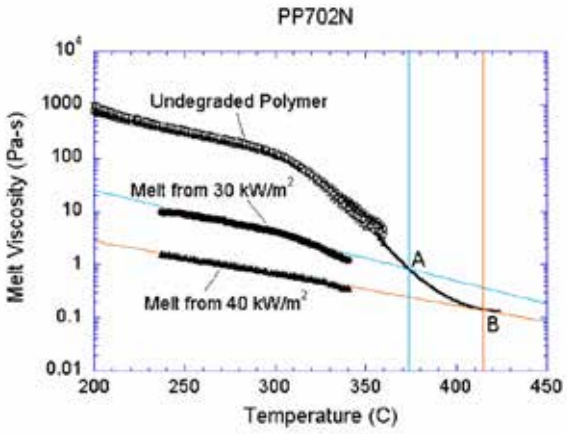


Figure 5.2: Viscosity of polypropylene as a function of temperature in its initial undegraded form and after exposure to 30 kW/m2 and 40 kW/m2 heat fluxes.

undegraded polymer the values is of 10E03, with a difference of several orders of magnitude.

The Particle Finite Element Method for simulation of the melting polymer Until the recent past there existed no methodologies whatsoever for approaching the polymer melting problem even without considering the interaction with the ambience. Attempts to model melt flow of polymeric material in fire using the Eulerian framework equipped with the volume of fluid (VOF) method have encountered difficulties with numerical instabilities and excessive runtimes [85]. On the other hand the application of the PFEM to the polymer melting simulation appeared to be very promising [82], [83].

Except for the general advantages of using PFEM for free surface flows, the method has several other benefits for the specific problem of interest. First of all, at the early stages when polymer temperature is relatively low and the viscosity is low, the deformations are small and Lagrangian description is as natural as that for the solid. At later stages when viscosity decreases drastically the flow propagates in bulk as well as in droplets. The capability of PFEM to simulate these flow discontinuities gives us a confidence in the feasibility of its applicability here [80], [84]. Finally, properties alteration at the surface due to the heating is dealt with automatically, since the properties are stored at the moving nodes. The evolving polymer surface "carries" its properties to a new position. In an Eulerian framework tackling of the changing properties at the evolving polymer

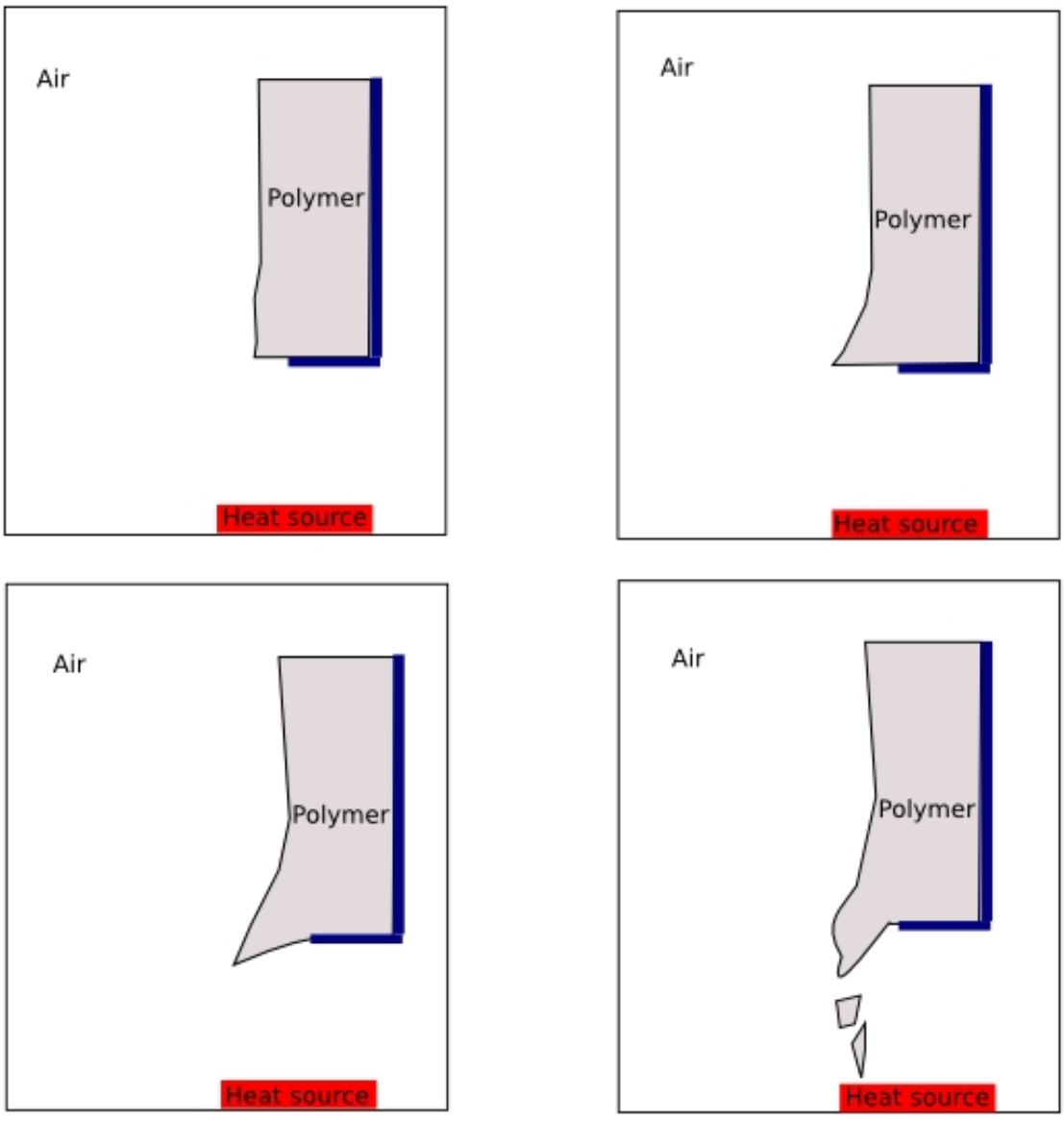


Figure 5.3: Free surface evolution of a polymer exposed to a heat flux

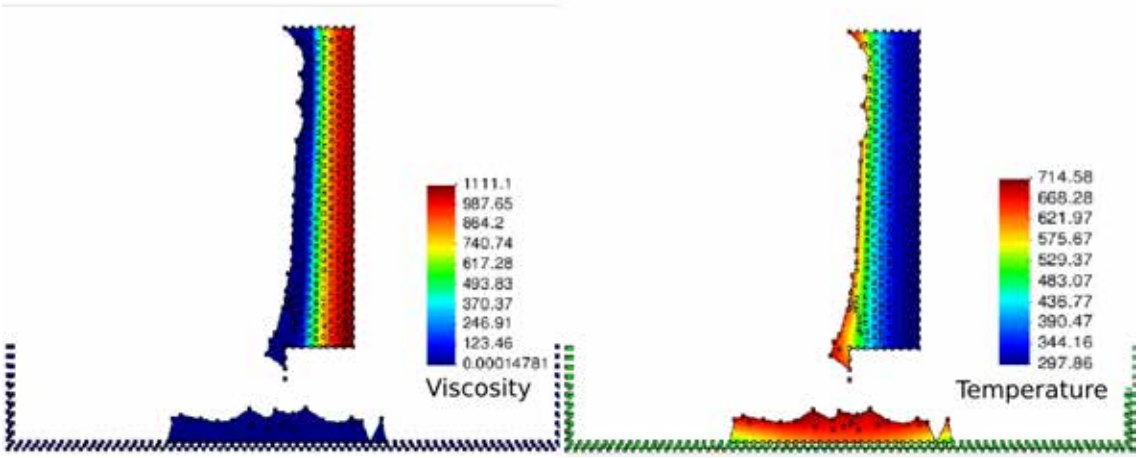


Figure 5.4: Viscosity and temperature distribution in a polymer heated with 20kW constant heat flux (t=700 s)

surface would require the solution of the transport equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0 \tag{5.1}$$

where ϕ is the property to be convected[1].

Out of different existing versions of PFEM (incompressible Lagrangian, incompressible ALE, quasi-incompressible Lagrangian, compressible Lagrangian) we choose the quasi-incompressible methodology presented in **Chapter 3**[2]. The density of the heated polymer practically does not change, thus enabling us to treat it as incompressible. The values of bulk modulus must assure only the absence of considerable mass loss, as the velocities encountered in the applications of interest are relatively low (maximum velocities are found in the dripping process due to gravity) , thus permitting the use of moderate values. We choose the value of bulk modulus of $\approx$ 10E6 [Pa], for which the convergence characteristics were found to be optimal (see **Chapter 3**).

The new feature to be included in the model of **Chapter 3** is the temporal and spatial dependence of the viscosity. The viscosity maps obtained using the empirical curves (Fig. 5.2) according to the

[1]Note that this problem occurs only if the fluid properties are changing, otherwise (e.g. for incompressible isothermal flow) only the distance function (in case of Level Set method) or the fraction function (in case of Volume of Fluid method) representing the free surface position needs to be convected.

[2]It is possible to use also the classical incompressible PFEM, presented in **Chapter 2**. However if in the future the interaction with flexible structures might be of interest, quasi-incompressible formulation has an advantage with respect to simplicity of the associated FSI strategy. Additionally the quasi-incompressible formulation has proven to preserve mass very well. The fractional-based incompressible Lagrangian or ALE formulations leads to certain difficulties in this respect. The monolithic single-field PFEM resolves this issue but is computationally expensive.

temperature distribution obtained by solving the thermal problem (the solution of the thermal problem are specified further below) will be provided as an input for the Lagrangian solver.

The PFEM module of the overall solver can then be summarized in the following algorithm:

1. **Input**: Temperature field (solution of thermal problem)
2. Obtain the viscosity distribution using the empirical relations (Graph 5.2)
3. Assemble the heterogeneous viscous matrix: $\mathbf{C}^{'} = \sum_{el}(\mu_G \int_{\Omega_{el}} \nabla\mathbf{N}\nabla\mathbf{N}\, d\Omega)$ where μ_G is the viscosity at the Gauss point of each element.
4. Solve the momentum/continuity system (3.15) (using $\mathbf{C}^{'}$) and (3.22) in the manner specified in **Chapter 3**
5. **Output**: New position of polymer and surface velocity

By **Input** we refer to the information passed to the current module (mechanical fluid solver in this case) by the other module (thermal solver in here) as the given data. By **Output** the data necessary to be passed to the other modules is indicated (in this case - to the solver for the air).

5.2.2 Model for the ambient air

The assumption of the constant heat flux over the whole polymer free surface used in [80], [82] is an over-simplification. In reality hot air is being convected inside of the control volume and therefore the heat flux distribution is not uniform neither in space nor in time. The physical phenomenon that is responsible for the air motion is mainly the natural convection, i.e. motion of air instigated by the density alteration due to heating. Therefore we propose to include the surrounding air into the model.

Before proceeding further we should make a comment upon the type of the coupling strategy we are aiming to develop as it strongly motivates the choice of the model for the air. Noticing that the ratio between the air and polymer density $\frac{\rho_a}{\rho_p} \propto 0.001$ as well as the fact that air velocities due to natural convection and the air pressure due to thermal expansion are also small[3], we propose to neglect in our model the mechanical action of air upon the polymer. Thus we aim at developing a one-way coupling and next address the kinematic description for the air in this scope.

[3]Low pressure values are responsible for neglecting the normal stress, and low velocities and viscosity of the air permit us to neglect the shear stress. Temperatures expected in the problem of interest are of order $\propto 500$ [C], which would result in a pressure increment (in case the ideal gas state equation is considered under isochoric conditions) of $\propto 6$ [MPa]

Keeping in mind that the polymer is modeled using the PFEM, we next consider the kinematic description for the air optimal for a weak coupling. There exist two main alternatives in terms of the frameworks choice for the ambient air:

1. Mesh-moving (Lagrangian or ALE)

2. Fixed mesh (Eulerian)

Even though the Lagrangian/ALE description is found to be advantageous in treating free surface flows, this is not the case for the ambient air, which is a vessel compliant flow. In the context of a multi-field problem only, the interface identification of the interface remains necessary. Hence a possible advantage in applying Lagrangian or ALE descriptions for the ambient air consists in the "exact tracking" of the air-polymer interface and "exact imposition" of the interface boundary conditions at common interface nodes, which is lost generally in case of a fixed Eulerian approach (if no additional techniques are used). Nevertheless, this precision is not crucial in a weak coupling, as no interface iterations are performed (similarly to classical weak couplings in aerodynamics).

In the perspective of the field of application of interest we can also notice that the air volume in the living areas is much larger than the total volume occupied by the furniture (polymer-made objects). Application of the Lagrangian/ALE description for the surrounding air would require re-meshing of the largest sub-domain, which would add unnecessary computationally cost without gaining practically anything. Thus for the problem of interest the possible advantages of the mesh-moving frameworks are minor in comparison with the additional computational cost due to re-meshing of the major part of volume.

Choosing Eulerian description for the air would permit us to avoid the re-meshing for the major part of the domain, thus leading to an efficient solution for the air. Using PFEM for the polymer facilitates the air-polymer interface identification. In fact it is exactly defined by the Lagrangian discretization, thus no additional techniques are required.

Consideration of the temperatures expected in the fire conditions requires utilization of a thermally coupled compressible formulation. For this purpose the sub-sonic compressible formulation developed in the **Chapter 4** will be used.

In Table 5.1 the justifications of the frameworks choice for both the subsystems are summarized.

In the following we specify the mechanical coupling strategy between the Lagrangian polymer and Eulerian air. Note that due to the fact that in the compressible Eulerian solver we shall use for the air, the energy equation solution is decoupled from the momentum/continuity system (for the polymer it

1. Lagrangian fluids are useful for gravitational flows involving free surfaces
2. Accuracy of ALE/Lagrangian descriptions in representation of the interface is necessary for the strong couplings
3. Eulerian fluids are standard and efficient for solution of the vessel compliant flows: shape functions etc. do not need to be recomputed
4. Lagrangian fluid defines the position of its domain boundary automatically
5. Having a Lagrangian polymer description, the position of the air-polymer interface is naturally defined, so no additional techniques need to be applied within the Eulerian mesh to detect the position of the polymer
6. In a room the melting object constitutes a minor part of the volume: re-meshing of the minor volume part would not affect the computational times considerably
7. Application of a Lagrangian method for the air would necessitate frequent re-meshing involving millions of elements where it really has no advantage

Table 5.1: Justification of the kinematic frameworks for modeling the air and polymer subsystems

is understood mutually as it is an incompressible). Thus we can address the mechanical and thermal solutions in separate blocks, which would not be the case if the energy-momentum/continuity system were coupled.

5.2.3 Mechanical coupling strategy

The coupling between the Eulerian and Lagrangian sub-domains falls into the class of "immersed boundary methods" [86], i.e. the Lagrangian sub-domain moves on top of the Eulerian mesh, forcing the Eulerian fluid (air in our case) to deform as the Lagrangian part [40]. The interaction is considered to take place on the Eulerian mesh between the Eulerian ambience and the boundary of the Lagrangian domain image (projection) onto Eulerian mesh. The background Eulerian mesh thus is separated into the "real" part, actually representing the air, and the "fictitious" part, that is occupied by the Lagrangian image. In this setting the interface position is exactly determined by the position of the Lagrangian part, and no additional techniques for the interface tracking are necessary. The fictitious domain (Lagrangian image) is defined by the curve/surface connecting the intersections of the Lagrangian surface with the background Eulerian mesh. Thus the precision of the interface definition depends exclusively on the mesh quality.

A coupling between the Eulerian and Lagrangian domains in the framework of the immersed boundary method consists in

- Mapping the Lagrangian part onto Eulerian mesh to obtain the Lagrangian image and the

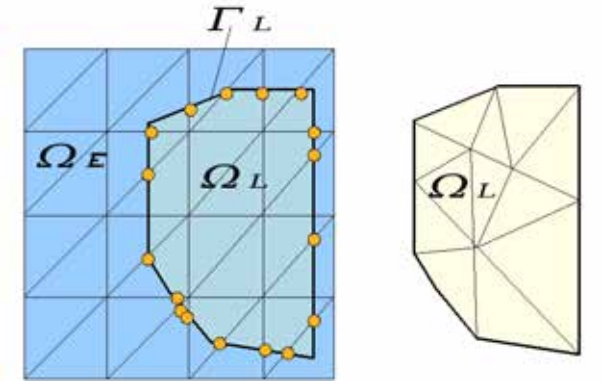


Figure 5.5: Setting: Eulerian and Lagrangian domains and their intersection

interface position, thus splitting the Eulerian domain into real and fictitious part

- Imposition of the interface boundary conditions (in general, Dirichlet-Neumann coupling):the polymer surface velocity as a Dirichlet condition for the air, and the forces (pressure+shear) exerted by the air as a Neumann condition for the polymer surface.

In case of the weak one-way coupling, the second constituent (Dirichlet-Neumann coupling) simplifies to exclusively Dirichlet coupling.

The mapping of the Lagrangian sub-domain onto the Eulerian mesh, namely the search for the intersections between the boundary of Lagrangian domain with the Eulerian mesh is facilitated using standard line intersections and spatial search techniques [87].

We directly concentrate on the imposition of the boundary conditions, assuming that the fictitious domain (Lagrangian image) is identified.

Imposition of the interface Dirichlet boundary condition Let us consider a domain Ω_e covered by a fixed Eulerian mesh. On top of this domain there exists a domain Ω_L, that is covered by a Lagrangian mesh. We denote the boundary of the Lagrangian domain by $\Gamma_L = \partial\Omega_L$. The setting is illustrated in Fig. 5.5. The intersections of Γ_L with the edges of Eulerian elements are denoted with the yellow circles. A polygon that connects these intersections splits the Eulerian domain into two sub-domains Ω_r and Ω_f, which correspond to the parts of Eulerian domain that lie outside and inside of the Lagrangian image, respectively (see Fig. 5.5). Indices "r" and "f" stand for "real" and "fictitious", since from the "point of view " of the Eulerian air, the domain corresponding to the interior of the Lagrangian domain has no physical meaning. (This is completely justified physically as the solution of the equation of the motion of the air does not depend on what happens inside the polymer.) Finally we introduce the term "interface Eulerian elements", that stand for those elements of the background mesh that are intersected by Γ_L.

The task to be carried out is to represent the velocity $\mathbf{v}_L$ of the Lagrangian boundary Γ_L within the Eulerian mesh and apply it as a boundary condition for the real part of the Eulerian domain.

There are several ways to do it. One consists in the identification of the intersections between Γ_L and Ω_E (yellow dots in Fig. 5.5) and splitting the intersected element into sub-elements, so that the intersection points coincide with the newly introduced nodes. The Dirichlet conditions can then be imposed in a "strong" way: exactly at the newly defined nodes. The drawback of the method is that it requires new degrees of freedom to be introduced for the Eulerian domain, as the interface evolves in time. This has a clear disadvantage, since the global discrete matrix requires then resizing and reconstruction at every time step, which noticeably slows down the computational speed. This is crucial for the simulation of the problem of interest since the real times to be simulated reach several minutes or even hours.

Another methodology is based on the ideas of Codina and Baiges [88]. The idea is to utilize the degrees of freedom associated to the nodes of the interface Eulerian elements that lie on the fictitious side of the domain (red dots in Fig. 5.6) and impose the boundary conditions in a weak sense (point-wise,"exact" imposition is impossible as long as the interface does not pass through the Eulerian nodes). The Dirichlet boundary condition that is representing the velocity field of the Lagrangian boundary will be further on called "projection Dirichlet boundary condition" to indicate that it comes from projecting the velocity field from the Lagrangian mesh onto the Eulerian mesh.

Given the velocity of the Lagrangian domain surface $\bar{\mathbf{v}}_L$ and the velocity field of the Eulerian real domain $\mathbf{v}$, a weak imposition of the boundary condition means the minimization of the difference between the velocity $\mathbf{v}$ and the given velocity (velocity of the Lagrangian part at intersection) in an integral sense[4], namely

$$\int_{\Gamma_L} \boldsymbol{\omega}(\mathbf{v}_h - \bar{\mathbf{v}}_L) = 0 \tag{5.3}$$

The elements, all nodes of which lie in the fictitious side are "switched off", i.e. excluded from the analysis. The nodes of interface elements, that lie on the fictitious side are used merely for the boundary condition imposition, and are present in the discretization of the real domain part. So finally the solution of the real part of the Eulerian domain $\mathbf{v}$ must on one hand satisfy the solution of the physical equations (Navier-Stokes), and on the other hand comply with the projection boundary conditions.

[4]This can be viewed as a solution of the minimization problem of a functional

$$\mathbf{I} = \int_{\Gamma_L} (\mathbf{v}_h - \bar{\mathbf{v}}_L)^2 \tag{5.2}$$

This permits to interpret $\mathbf{v}_h$ as an optimal solution of the problem in the least square sense.

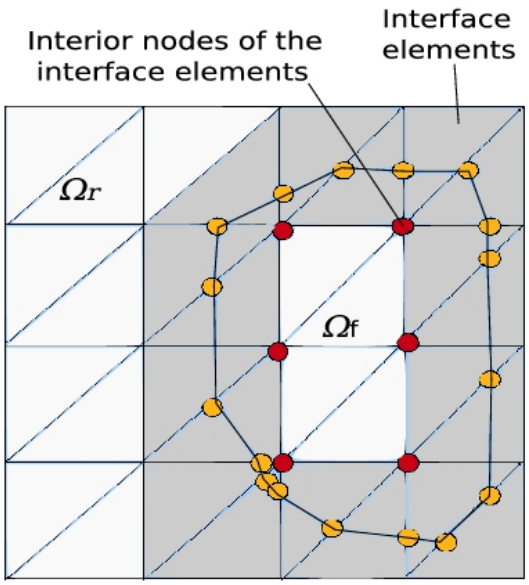


Figure 5.6: Setting for the approximate imposition of the projection Dirichlet b.c.

Suppose that the unknown velocity field $\mathbf{v}$ can be expressed as:

$$\bar{\mathbf{v}}(\mathbf{x}) = \sum_{i=1}^{n_r} N_r^i(\mathbf{x})\mathbf{v}_r^i + \sum_{j=1}^{n_f} N_f^j(\mathbf{x})\mathbf{v}_f^j \tag{5.4}$$

where N_r^i and N_f^j are the standard interpolation function, n_r are the number of nodes in Ω_r (outside of the Lagrangian image) and n_f is the number of nodes of the interface Eulerian elements lying on the fictitious side. The variable corresponding to the real part of domain is denoted as $\bar{\mathbf{v}}_r$ and the one corresponding to the fictitious part of the interface elements as $\bar{\mathbf{v}}_f$.

The governing discrete system for the real part of the domain can be written as

$$\mathbf{A}\bar{\mathbf{v}} = \mathbf{F}_r \tag{5.5}$$

where $\mathbf{A}$ includes all the discrete matrices of Navier-Stokes equations and $\mathbf{F}$ is the load vector. Note that for the interpolation we are using also the nodes of the interface Eulerian elements lying within the fictitious domain, hence Eq. (5.5) can be rewritten as

$$\mathbf{A}_{r\,r}\mathbf{v}_r + \mathbf{A}_{r\,f}\mathbf{v}_f = \mathbf{F}_r \tag{5.6}$$

On the other hand, Eq. (5.3) has to be satisfied. This gives

$$\mathbf{M}^a_{\Gamma_L}\mathbf{v}_r + \mathbf{M}^b_{\Gamma_L}\mathbf{v}_f = \mathbf{f}_{\Gamma_L} \tag{5.7}$$

where $\mathbf{M}^a_{\Gamma_L} = \int_{\Gamma_L} \mathbf{N}_r^T\mathbf{N}_r d\Gamma_L$, $\mathbf{M}^b_{\Gamma_L} = \int_{\Gamma_L} \mathbf{N}_f^T\mathbf{N}_r d\Gamma_L$ and $\mathbf{f}_{\Gamma_L} = \int_{\Gamma_L} \mathbf{N}_f^T\mathbf{N}_r d\Gamma_L$.

Eqs. (5.6) and (5.7) are the system of equations that both include the governing equations of motion of the real part of the Eulerian domain and the equations that correspond to the weak imposition of the projection Dirichlet boundary conditions. As we see, the solution is permitted to "adjust" in such a way as to satisfy the projection Dirichlet boundary conditions by using the degrees of freedom of the interface Eulerian elements of the fictitious part.

A first option in resolving the problem consists in the simultaneous solution of the system (Eqs. (5.6) and (5.7)):

$$\begin{pmatrix} \mathbf{A}_{r\,r} & \mathbf{A}_{r\,f} \\ \mathbf{M}^a_{\Gamma_L} & \mathbf{M}^b_{\Gamma_L} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{v}}_r \\ \bar{\mathbf{v}}_f \end{pmatrix} = \begin{pmatrix} \mathbf{F}_r \\ \mathbf{f}_{\Gamma_L} \end{pmatrix} \tag{5.8}$$

Also it is possible to express the velocities corresponding to the fictitious nodes from Eq. (5.7) and substitute them into (5.6) to give

$$\left[\mathbf{A}_{rr} - \mathbf{A}_{ff}\mathbf{M}^{-1}_{\Gamma_L}\mathbf{M}^b_{\Gamma_L}\right]\bar{\mathbf{v}}_r = \mathbf{F}_r - \mathbf{M}^{-1}_{\Gamma_L}\mathbf{f}_{\Gamma_L} \tag{5.9}$$

However this would imply modification of the left-hand-side of the discrete system (subtraction of the term $\mathbf{A}_{ff}\mathbf{M}^{-1}_{\Gamma_L}\mathbf{M}^b_{\Gamma_L}$), which would preclude an efficient implementation in the explicit context[5].

Another option is to solve Eq. (5.7) in a staggered way, i.e. at time step $n+1$ obtain $\bar{\mathbf{v}}_f^{n+1}$ as

$$\bar{\mathbf{v}}_f^{n+1} = \mathbf{M}^{b-1}_{\Gamma_L}\mathbf{f}_{\Gamma_L} - \mathbf{M}^{b-1}_{\Gamma_L}\mathbf{M}^a_{\Gamma_L}\bar{\mathbf{v}}_r^n \tag{5.10}$$

and then plug it in the Eq. (5.6). This implies that instead of solving for the velocity $\bar{\mathbf{v}}_{n+1}$, that satisfies both the physical equations of the "real" fluid part and the projection boundary conditions, we proceed as follows:

1. Given $\bar{\mathbf{v}}_r^n$ we solve the system (that is assembled from exclusively interface elements contributions), for the velocity of the fictitious nodes of the interface elements $\bar{\mathbf{v}}_f^{n+1}$, according to Eq. (5.10).

2. Fix the newly computed velocity at the fictitious nodes (application of the projection Dirichlet

[5] Unless the term $\mathbf{A}_{ff}\mathbf{M}^{-1}_{\Gamma_L}\mathbf{M}^b_{\Gamma_L}$ is lumped

boundary conditions)

3. Solve the equations of the fluid domain, assembled from the real and interface elements contributions.

The approximation of this methodology originates from the fact that the fictitious nodes velocity $\bar{\mathbf{v}}_f$ is computed considering the real domain velocity $\bar{\mathbf{v}}_r$ from the previous time step t_n, instead of t_{n+1}. This is a first order prediction and it is justified when small time steps are used[6].

We follow the latter option as it does not require modification of the mass matrix which is crucial for the explicit algorithm used in the air solver. Additionally it permits to define a new module for the interface velocity calculation, leaving the Navier-Stokes solver unchanged. Preserving the modular approach consists in the definition of additional building blocks, while keeping the existing ones as general as possible.

The solution algorithm for the ambient air part of the solver can be summarized as follows:

1. **Input**: Position of the Lagrangian sub-domain within the Eulerian mesh; Lagrangian surface velocity $\bar{\mathbf{v}}_L$; temperature distribution $\bar{T}_{n+1}$ (solution of the thermal module. To be specified further)

2. Solve the interface projection equation, minimizing the difference between $\bar{\mathbf{v}}_L$ and $\bar{\mathbf{v}}_n$ on the interface elements. Obtain the solution $\bar{\mathbf{v}}_f$

3. Fix the velocity at the fictitious nodes of the interface elements to $\bar{\mathbf{v}}_f$

4. Solve the flow equations (compressible Navier-Stokes momentum/continuity system)

5. Output $\bar{\mathbf{v}}_{n+1}$, $\bar{p}_{n+1}$, $\bar{\rho}_{n+1}$

Remark It is very simple and straight-forward to extend the proposed methodology to a two-way Dirichlet-Neumann coupling. In fact it simply requires mapping of the air pressure field onto the polymer surface, and its application as the Neumann boundary condition for the Lagrangian (polymer) solution step. Note that it would be convenient to neglect the frictional action because of the poor resolution in the vicinity of the wall.

Next we address the final ingredient of the coupling strategy, namely the energy equation solver.

[6]Of course it is possible to plant this procedure in an iterative context. However multiple solution of the momentum/continuity system is computationally inefficient.

5.2.4 Thermal interaction

We propose to solve the thermal problem in a monolithic way on the Eulerian mesh. The governing equations for both subproblems are exactly the same, the only difference being connected to the thermal properties: specific heat and conductivity. The idea consists in representing the Lagrangian domain within Eulerian mesh by mapping the Lagrangian domain onto the Eulerian one and assigning to the Lagrangian image the corresponding thermal properties of the polymer. The solution of the thermal problem is then performed on the Eulerian mesh, that consists of (see Fig. 5.7).

- elements with the properties of air (the ones that lie outside of the Lagrangian image)
- the elements with the properties of polymer
- the "transition" elements that correspond to the ones that are crossed by the Lagrangian image[7]

The convective velocity inside of the domain part corresponding to the polymer is fixed to zero, whereas in the air convective velocity is obtained by solving the equations of motion. Once the thermal problem is solved on the Eulerian mesh, the solution (temperature field) has to be transferred back to the Lagrangian mesh.

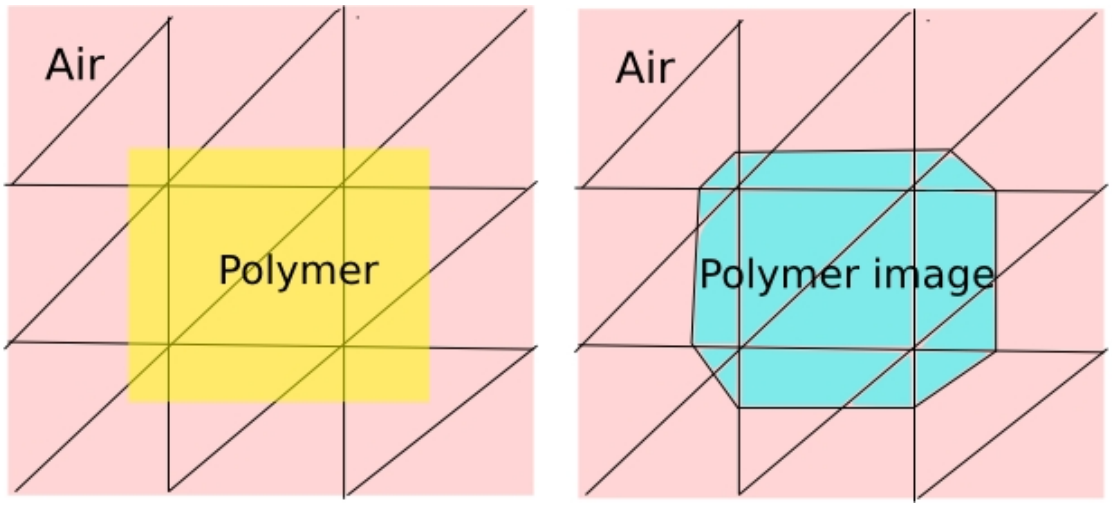


Figure 5.7: The image of the Lagrangian part on the Eulerian mesh

[7]note that in contrast with the mechanical coupling, here we do not switch-off any part of the Eulerian domain

Data transfer Next we shall present a methodology for transfer of the data between the Lagrangian and Eulerian mesh. The methodology can be however applied in general for data transfer of a scalar or vector variable z between non-matching meshes. To present the methodology in the general context we shall use further on the terms "origin" and "destination" meshes instead of Lagrangian and Eulerian ones, keeping in mind that in case of thermal properties projection the "origin" is the Lagrangian mesh, but in case of the solution (temperature) projection the origin is the Eulerian mesh. Note that the location of the Lagrangian image on the Eulerian mesh changes at every time step of the transient problem and thus the location of the "polymer properties" within the Eulerian domain of the thermal problem has to be identified at every time step.

Obviously, point-wise equality of the variable of interest on the non-matching meshes cannot be achieved. There exist various techniques for the data transfer; we shall present two of them.

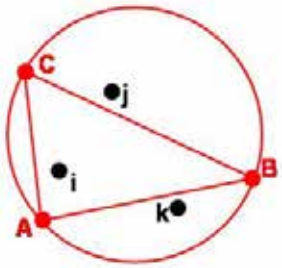


Figure 5.8: Data transfer: direct interpolation

Direct interpolation One possibility for the transmission of data between non-matching meshes is the direct interpolation between the nodal values of the origin and the destination mesh. Given an origin mesh and a destination mesh, for each node of the destination mesh, we find out which element of the origin mesh it is contained in. Once the element is identified, a simple finite element interpolation with the shape functions of the element of the origin model part is performed. Finally the value of the variable in the new point is given. Once we found that only node i of Fig. 5.8 (note that ABC are nodes of the origin model part and ijk are of the destination model part) is inside element ABC, we can calculate the value of variable q simply through FE interpolation: $q_i = N_A(x_i)q_A + N_B(x_i)q_B + N_C(x_i)q_C$. In this simplest method only the spatial search of the origin mesh elements containing the given point i needs to be implemented efficiently.

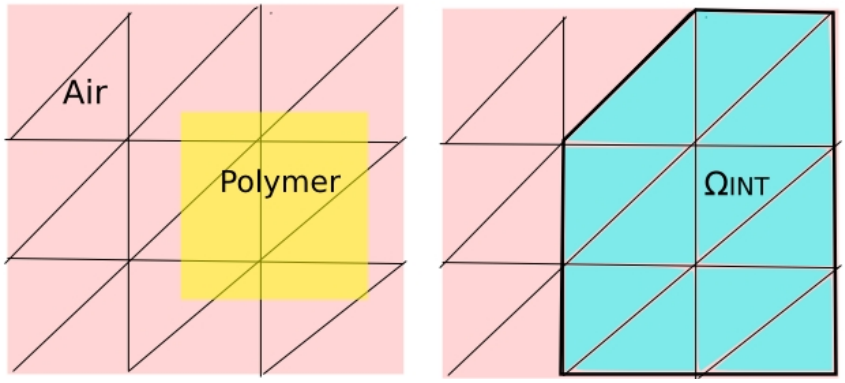


Figure 5.9: Domain of integration for the minimization problem orginiating from the projection

L2-projection Another method consists in satisfying the equality of the variable of interest on both meshes in a weak sense[8]. Equality of the the variable of interest z between destination and origin over the interface in an integral sense can be expressed as follows (method of weighted residuals, which coincides with the L2-projection of the origin field onto the destination one):

$$\int_{\Omega_{INT}} w(z_d - z_o) = 0 \tag{5.11}$$

and Ω_{INT} stands for the set of the elements of destination mesh (Eulerian) intersected by the origin ones (Lagrangian) and the ones that lie in the interior of the Lagrangian image. This is depicted in the Fig. 5.9. Since the integration is performed on the destination mesh, s is the test function, corresponding to the destination elements, i.e. $s = N_i^d q$

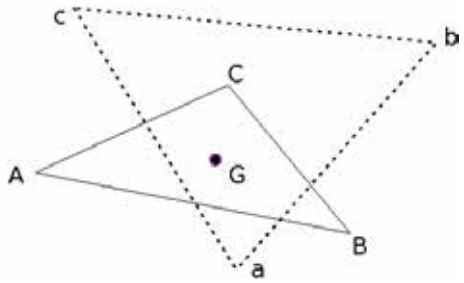


Figure 5.10: Origin (ABC) and destination (abc) elements, L2 projection

Taking the destination element ABC and origin element abc (see Fig. 5.10) we find the value

[8]similar to the technique we addressed for the weak imposition of the boundary conditions in the mechanical coupling

($z(G) = \sum_i N_i^o(Q) z_i^o$) of the origin element, corresponding to the coordinates of the Gauss point of the destination G. We are looking for such nodal values of the destination element nodes z_i^d ($i = A_d, B_d, C_d$), that

$$\int_{ABC} N_i^d z_i^d = \int_{ABC} N_i^d z_i^o \tag{5.12}$$

note that the integration is done over the destination mesh. The system to solve then can be written as:

$$\mathbf{M}_c \cdot z_d^i = \mathbf{f} \tag{5.13}$$

with $\mathbf{M}_c = \int_{\Gamma_d} N_d^i N_d^j$ being the "consistent mass matrix", and $\mathbf{f} = \int_{\Gamma_d} N_d^i N_o^j z_o^j$ the "force vector". The force vector $\mathbf{f}$ is obtained by integrating $N_d z_o(x) = N_d N_o z_o(x)$ on the destination domain. By interpolation with the shape functions of the destination element we get a distribution of the variable on the destination mesh. The result is the approximation of the distribution of the origin domain function $z(x)$ on the destination mesh, that guarantees the least square approximation.

Remark: discontinuity representation in case of the proposed methods of data transfer
An erroneous aspect of both the proposed methodologies (simple interpolation and the L2-projection) is the way they represent the discontinuity of the mapped variable at the air-polymer boundary. In case of the temperature mapping this is not the case, as temperature is continuous at the air-polymer boundary, however the conductivity and heat capacity are discontinuous. The discontinuity representation will be illustrated using a one-dimensional example for the sake of clarity. Fig. 5.11 shows the 1D Eulerian mesh (shown in black) and the Lagrangian mesh corresponding to the polymer (shown in yellow). Exact representation of the discontinuity of the thermal properties is the step function adopting the value ϕ_o (of the polymer) at the part of the domain corresponding to the polymer and the value ϕ_d elsewhere (property of the air). In case of the interpolation technique, the nodes of the Eulerian mesh that lie outside the Lagrangian domain take the property of the air, and the ones that lie within - the property of the polymer. Within the Eulerian elements that are crossed the distribution is linear. In case of a projection, both nodal values (that of the node that lies within the Lagrangian domain and the one that lies outside) change. Every Eulerian element whose Gauss point lies within Lagrangian domain contributes the projected variable values to all its nodes. This can be seen in Fig. 5.12.

The L2 projection method is more accurate is it aims to minimize the difference in the distribution of the variable of interest on both meshes in the least square sense. If the variable to be mapped is constant (as specific heat and conductivity in our case), the simple interpolation is more favorable as

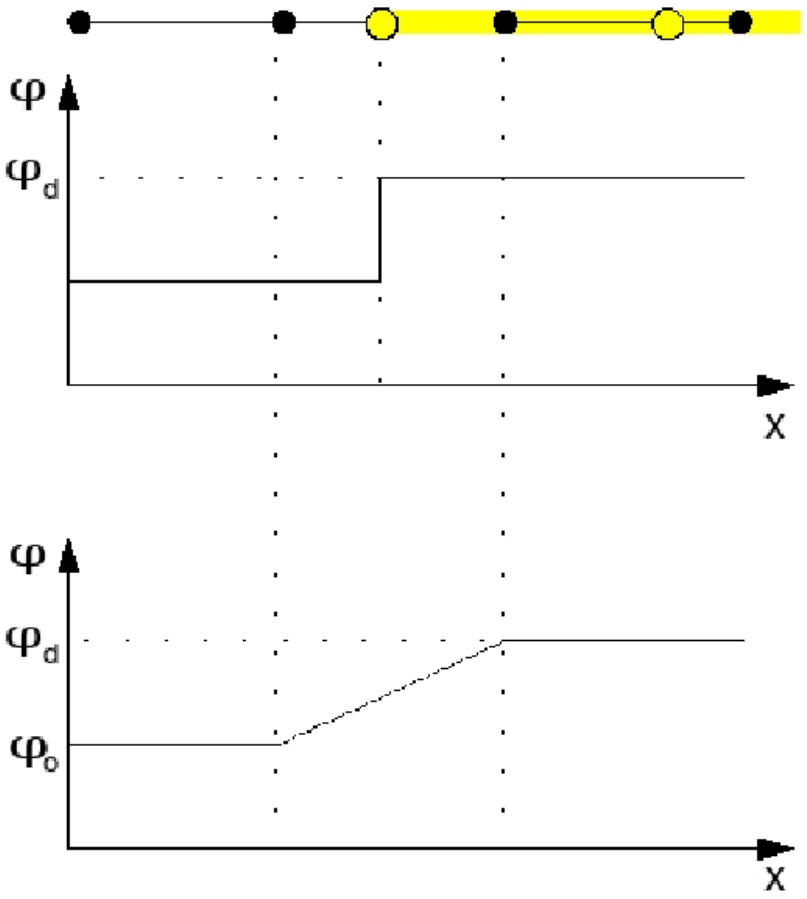


Figure 5.11: Representation of discontinuity in case of nodal interpolation

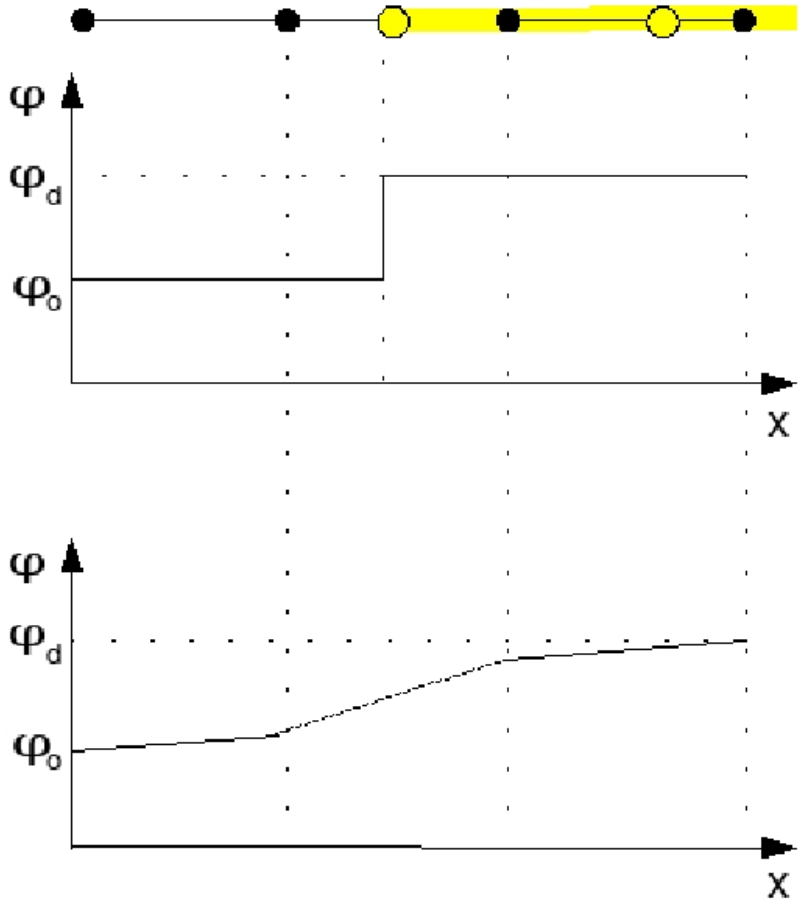


Figure 5.12: Representation of discontinuity in case of nodal interpolation

for the origin elements that lie completely within the Lagrangian image both methods give exactly the same result. The temperature field does not exhibit discontinuities across the air-polymer interface. Thus the temperature distribution from the Eulerian mesh onto the Lagrangian one should be mapped using the L2 projection.

Having described the different aspects that need to be considered when solving the thermal problem on the Eulerian mesh, representing both the air and the polymer, we can summarize the solution strategy. This is given in the Table 5.2

The overall solution algorithm At this point all the ingredients for the overall solution strategy are described. Before presenting the final algorithm we summarize the techniques used for each module:

1. Solution of the equations of motion for the air: compressible Eulerian solver

2. Solution of the equations of motion for the polymer: quasi-incompressible Lagrangian solver

1. **Input**: position, heat capacity, conductivity and temperature of the polymer
2. Re-set the thermal properties of the Eulerian mesh to those of air
3. Map polymer heat capacity, conductivity and temperature onto the Lagrangian image within the Eulerian mesh
4. Solve the energy equation on the Eulerian mesh
5. Map the obtained temperature distribution onto the Lagrangian mesh
6. **Output**: temperature distribution of the air and that of the polymer

Table 5.2: Algorithm for a monolithic solution of the thermal problem involving air and polymer on the Eulerian mesh

3. Solution of the thermal problem involving both the subsystems: Eulerian energy solver, monolithic solution for both the air and the polymer

4. Mechanical interaction between the polymer and the air: one-way coupling with projection Dirichlet b.c. imposed in a weak manner

The overall solution algorithm is presented in Table 5.3:

5.3 Examples

In this section we shall present the examples showing the functionality of the coupling. We first present several examples for validation of the mechanical coupling, then an overall thermo-mechanical coupling and finally conclude the section with a simulation of the polymer melting under unsteady heat flux.

5.3.1 Mechanical coupling validation

Weak boundary condition imposition The very first example is chosen in order to test the weak imposition of the interface boundary condition. A square domain (1x1 [m]) shown in the Fig. 5.13 is modeled using the Eulerian formulation. Left wall has a horizontal velocity $v_x = 1$ [m/s] prescribed, the rest of the walls are fixed. In the middle of the domain we locate a cylinder, where zero velocity is to be imposed. The whole problem is solved on the Eulerian mesh and our task is the representation of the fixed cylinder within the Eulerian mesh. The meshes for both parts (see Fig. 5.14) are chosen to be very rough in order to have a better qualitative insight.

Fig. 5.14 (a) shows the actual meshes used for the computation. We see that the Eulerian mesh

Table 5.3: Algorithm for the solution of thermo-mechanical problem involving polymer and the ambient air

1. Solve equations of motion for the polymer
   Input: polymer viscosity distribution
   Output: new position, velocity
2. obtain the image of the Lagrangian part on the Eulerian mesh
3. Map the heat capacity, conductivity and temperature of polymer onto Eulerian mesh (interpolation)
4. Solve the energy equation on the Eulerian mesh
   Input: air velocity, thermal properties of air and of polymer (Lagrangian image)
   Output: temperature distribution
5. Map the temperature distribution onto the Lagrangian mesh using L2 projection
6. Update the polymer viscosity distribution using the empirical viscosity-temperature relation
7. Solve the equations of motion of the ambient air
   Input: temperature distribution, polymer position, polymer surface velocity
   a) switch on all the Eulerian elements
   b) switch off all the Eulerian elements that lie entirely inside the fictitious part of the Eulerian domain)
   c) Apply the projection Dirichlet b.c. at the "fictitious" nodes of the Eulerian interface elements
   Output: air velocity
8. go to next time step

elements that lie completely inside of the image of the cylinder are eliminated from the calculation. The fictitious nodes of the interface elements are marked with the $INTERFACE$ marker. These nodes are the ones that are used for the imposition of the projection boundary condition.

Fig. 5.15 b) shows the iso-contour of zero velocity. One can see that the projection boundary condition is applied correctly, i.e. the velocity at the fictitious nodes is such that the velocity at the intersection between the Eulerian background mesh and the image of the cylinder is equal (in the weak sense) to the velocity of the cylinder (zero in this case).

Exact vs. approximate imposition of the boundary conditions Next example shall provide us an insight with respect to the quality of the mechanical coupling based upon an approximate imposition of the boundary conditions. The flow over a cylinder is chosen in the flow regime with $Re = 800$, where vortex shedding is expected. The velocity is set to $v_x = 2$ [m/s] at the inflow, the upper and lower walls have the slip condition prescribed ($v_y = 0$) and zero pressure is prescribed

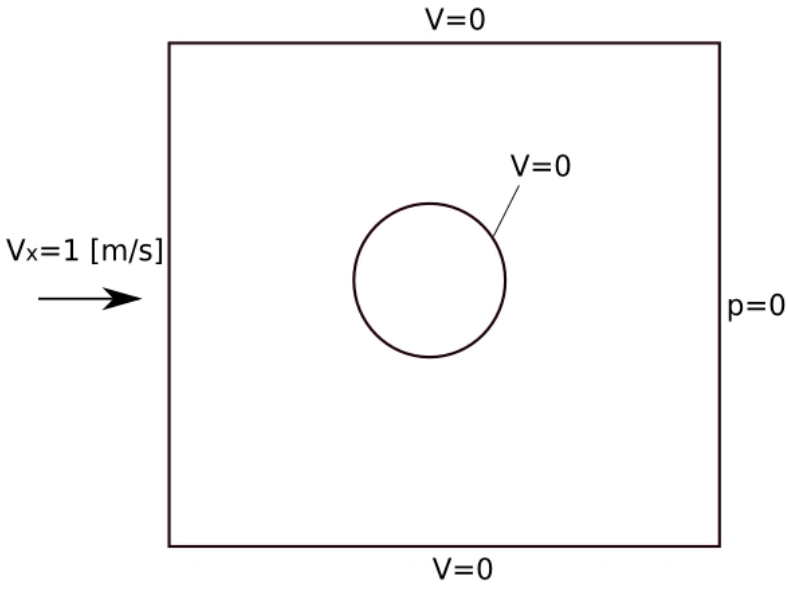


Figure 5.13: Weak imposition of projection b.c.: model

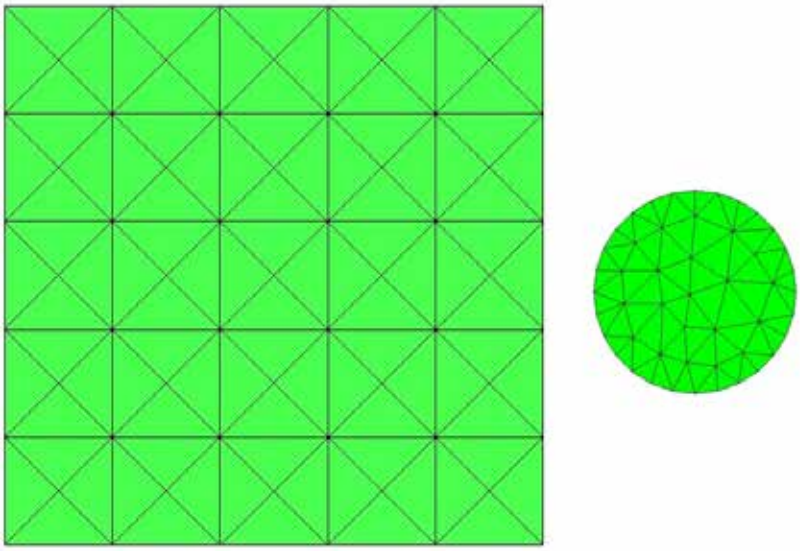
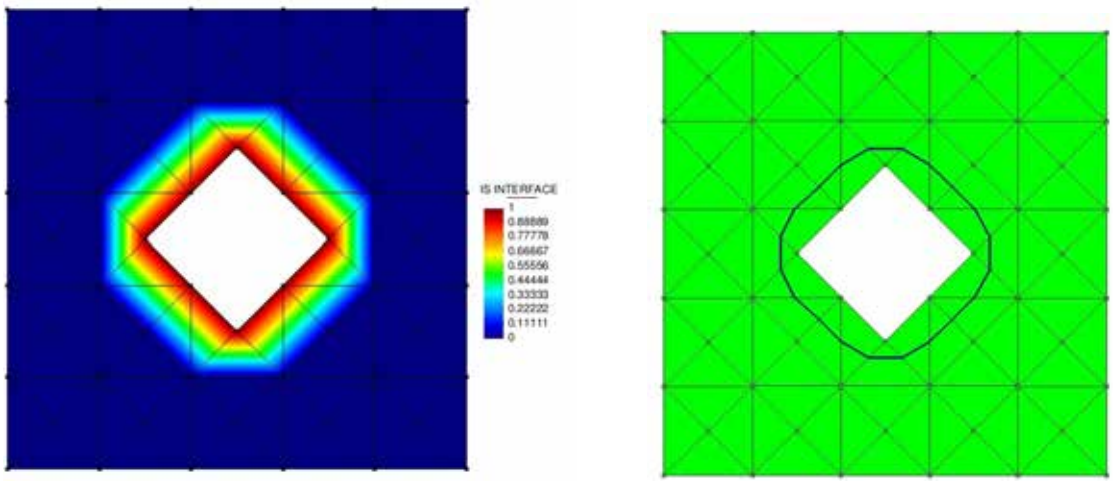


Figure 5.14: Weak imposition of projection b.c.: meshes

at the outflow. The computational domain has the size 1x0.2 [m], and a cylinder with radius 0.02 [m] is located in the middle (0.5, 0.1) of the rectangular domain. The example is discretized with ≈ 20000 linear triangles.

Fig. 5.16 and 5.17 show the velocity and pressure fields at time instances $t = 0.87$ and $t = 0.92$ when the flow is fully developed. The main period of velocity oscillation is $\approx 0.1s$, so the chosen time instances correspond to opposite phases. A comparison between the standard Eulerian solution with



(a) Actual computational mesh and interface elements

(b) Iso-contour of zero velocity

Figure 5.15: Projection boundary conditions imposition

exact imposition of boundary conditions[9] shows good agreement both for the velocity and pressure values.

A magnified image of the area around the cilinder in the case of the approximate imposition of boundary conditions is shown in the Fig. 5.18. The inner-most layer of nodes correspond to the fictitious nodes of the interface elements.

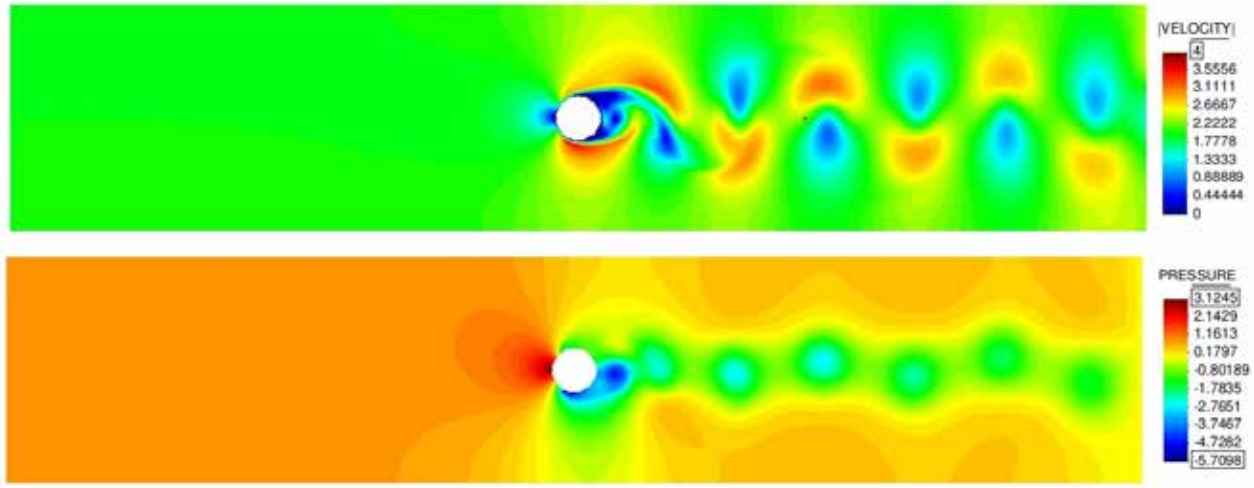
Falling object In the preceding examples the coupling was tested in situations with the object fixed. Next we present a simple example, showing the motion of the object in the closed area. The object exposed to gravity is falling in the ambient air.

The example is validated by comparing the velocity field with the solution obtained using a monolithic method developed in **Chapter 2**. The properties are summarized in the next table:

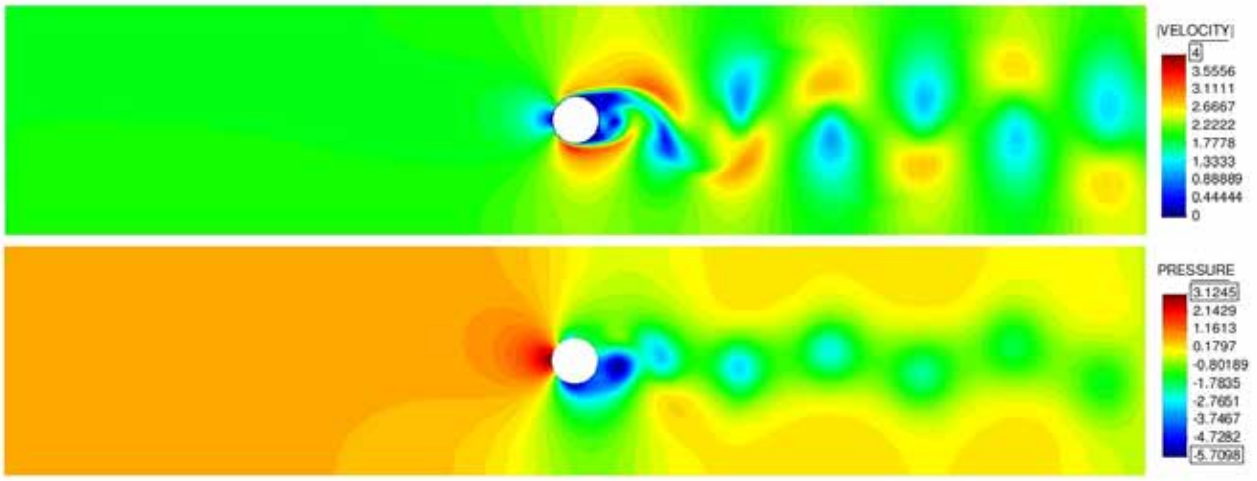
- Ambient air density $\rho_a = 1\ [kg/m^3]$
- Ambient air viscosity $\nu_a = 0.001[Pas]$
- Gravity acceleration $g_y = -10[m/s^2]$

Fig.5.19 displays comparisons of the velocity fields obtained by the weak coupling presented in the current chapter and with the monolithic strong coupling solution at two time instances.

[9] where the void corresponding to the cylinder is present in the model prior to meshing and thus the zero-velocity condition is exactly imposed on the nodes of the discretization corresponding to the cylinder contour

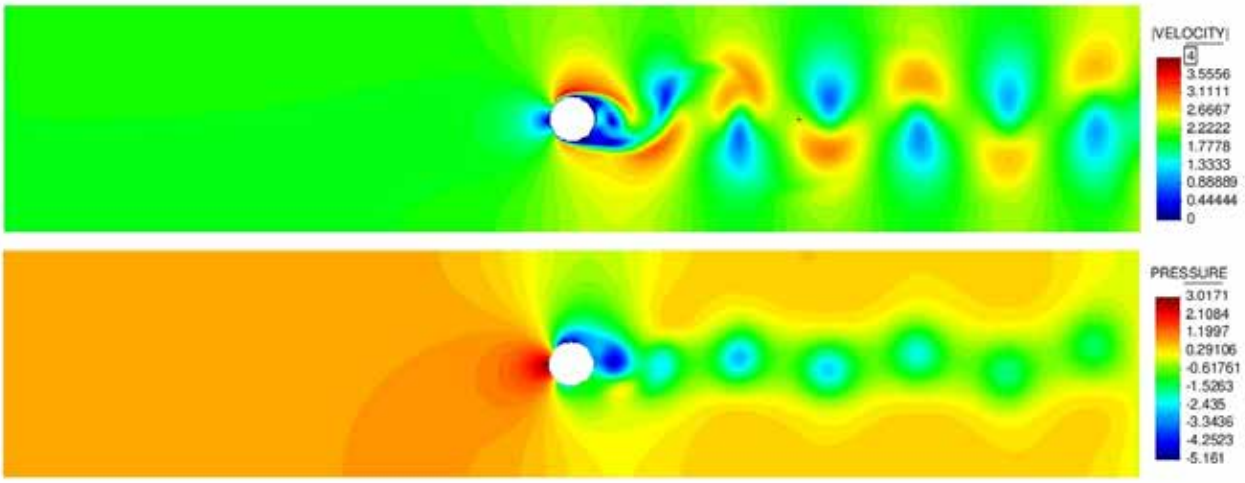


(a) Velocity and pressure: approximate b.c.

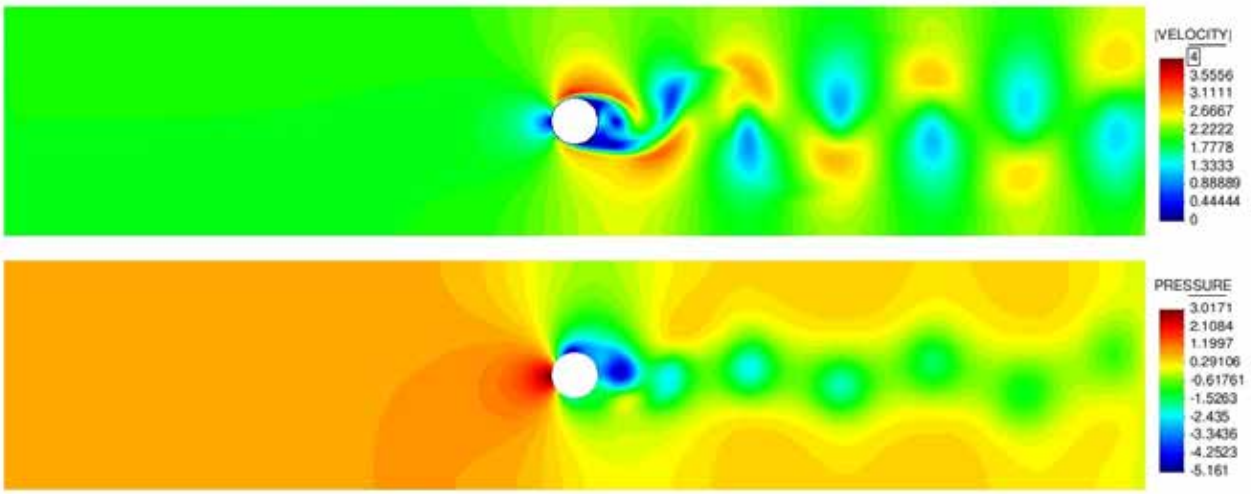


(b) Velocity and pressure: exact b.c.

Figure 5.16: Flow over cilinder: approximate vs. exact imposition of b.c., t=0.87 [s]



(a) Velocity and pressure: approximate b.c.



(b) Velocity and pressure: exact b.c.

Figure 5.17: Flow over cilinder: approximate vs. exact imposition of b.c., t=0.92 [s]

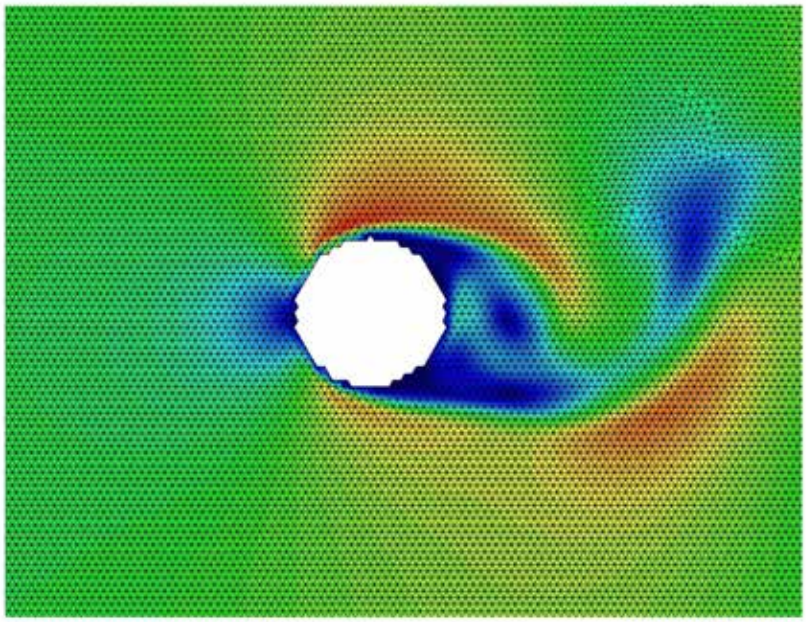
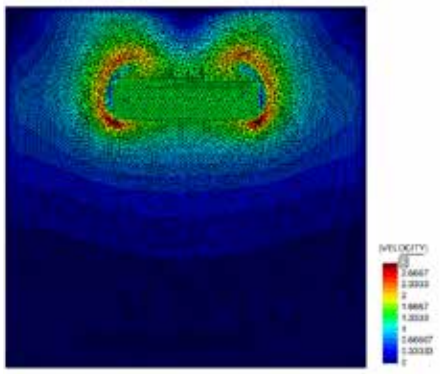
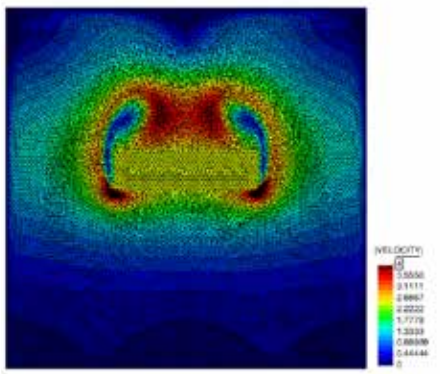


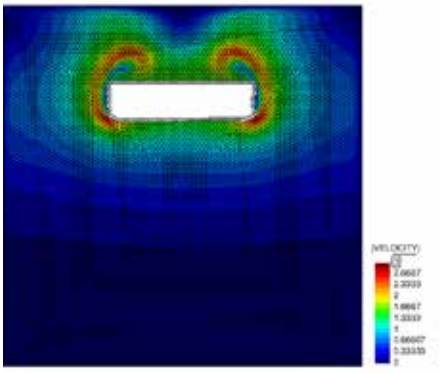
Figure 5.18: Magnified image of the area around the cilinder



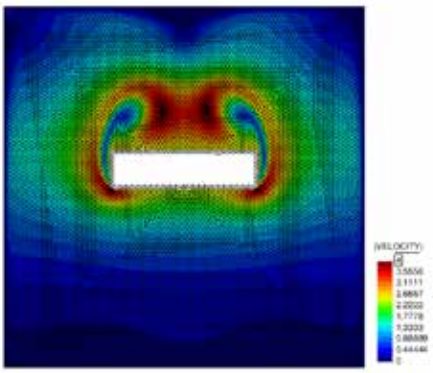
(a) Velocity at t=0.15 s: monolithic solution



(b) Velocity at t=0.25 s: monolithic solution



(c) Velocity at t=0.15 s: weak coupling



(d) Velocity at t=0.25 s: weak coupling

Figure 5.19: Comparison of results of a weak Eulerian-Lagrangian coupling strategy with the monolithic Lagrangian solution

One can see very similar velocity patterns with maximum values at the leading corners of the polymer, in both cases around 3 [m/s] (at t=0.15 [s]) and 4 [m/s] at t=0.25 [s].

Next we provide a quantitative comparison. Fig. 5.20 shows line graphs of velocity along two cuts: one is a horizontal cut at y=0.75, the second one is a vertical cut at x=0.5. (the location of cuts are shown on the right of the graphs)

We can see very good agreement between the graphs. Note that for the Eulerian-Lagrangian coupling strategy, the Lagrangian image domain is excluded from the Eulerian mesh. This explains the discontinuity in the graph corresponding to the Eulerian-Lagrangian coupling.

This example shows that the action of air upon the object neglected in the weak coupling does not lead to considerable error in the prediction of the velocity field. The air velocity profiles as well as the position of the polymer show very good agreement, providing confidence that the weak coupling is working correctly and that the simplifying assumption is acceptable.

5.3.2 Thermo-mechanical coupling validation

Here we present an example for validating the entire (thermo-mechanical) coupling. We choose the heated cavity example, similar to the one addressed in **Chapter 4**. The properties of the heated cavity domain are as follows:

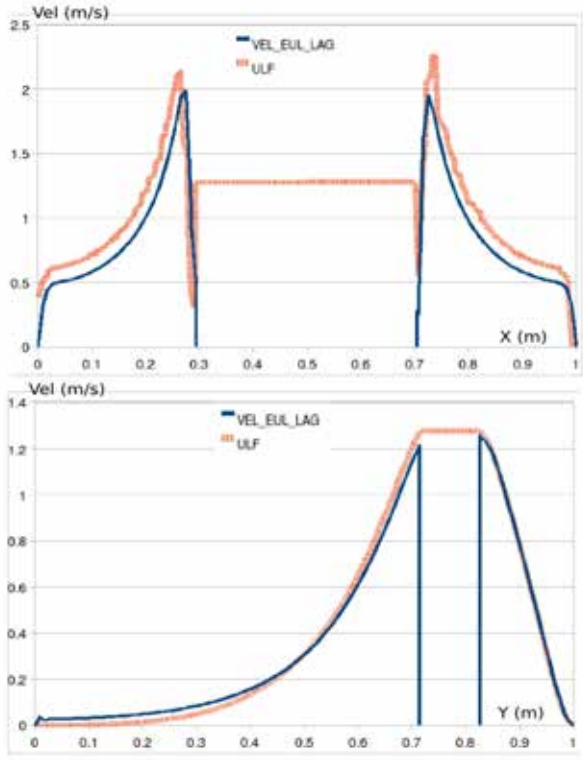
- heat capacity 1000 $[J/kg \cdot K]$
- conductivity 25 $[W/K \cdot m]$
- viscosity 0.001 $[Pa \cdot s]$
- initial temperature 300 $[K]$
- density 1 $[kg/m^3]$
- gravity 10 $[m/s^2]$

At the vertical walls of the cavity the temperatures of 305 [K] and 295 [K] are prescribed.

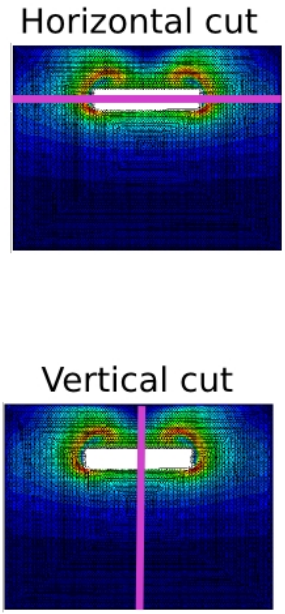
Here additionally a circular object, placed at $t = 0$ at the center of square domain (see Fig. 5.21) is subjected to a prescribed motion:

$$\mathbf{u} = 0.05 \cdot sin(5 \cdot t) \tag{5.14}$$

The circular object has following thermal properties:



(a) Velocity profiles



(b) Location of the cuts

Figure 5.20: Velocity profiles obtained using the monolithic updated Lagrangian method and partitioned Eulerian-Lagrangian coupling strategy

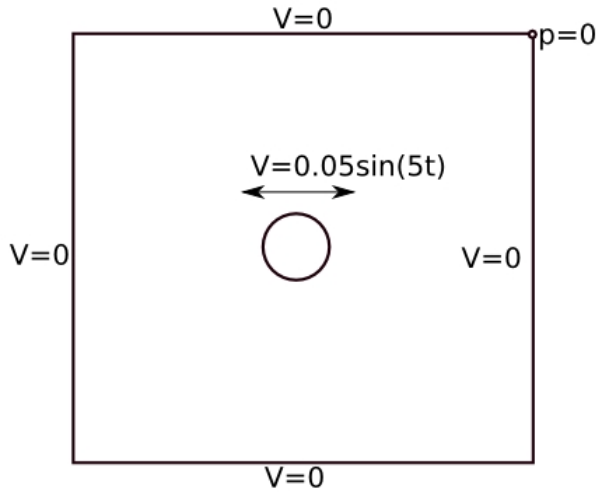


Figure 5.21: A heated cavity with moving circle: geometry

- heat capacity 10 $[J/kg \cdot K]$
- conductivity 1000 $[W/K \cdot m]$

Such settings are chosen in order to emphasize the thermal properties difference between the ambience and the object, thus facilitating the validation. The air domain was discretized with $\approx$ 20000 linear triangles.

We next compare the solution obtained using the Eulerian-Lagrangian strategy derived here with the one obtained using body-fitting ALE approach incompressible fractional-step based formulation (OSS-stabilized). In both cases Boussinesq modification of gravity term is taken into account.

Fig. 5.22 and 5.23 show the temperature and velocity contours obtained by both methods at time instances $t = 1.4[s]$ and $t = 15[s]$, corresponding to transitory stage and the periodic stage, when the temperature field is fully developed and the changes in the velocity field occur primarily due to the moving object.

One can see a good agreement between the corresponding distributions.

Finally we present the temperature evolution in time at the center of the moving cilinder. This shall enable us to test the thermal part of the coupling. On the Fig. 5.24 one can see that the behavior is similar, however the cycle with minimal amplitude is reached in the case of Eulerian-Lagrangian solver at $\approx$ 6.8 [s] while in the case of ALE solution it happens at $\approx$ 8.2 [s]. In fact the difference between the two graphs consists precisely in one additional cycle with high amplitude in the case of the ALE solver. We believe that this difference can be explained by the fact that the

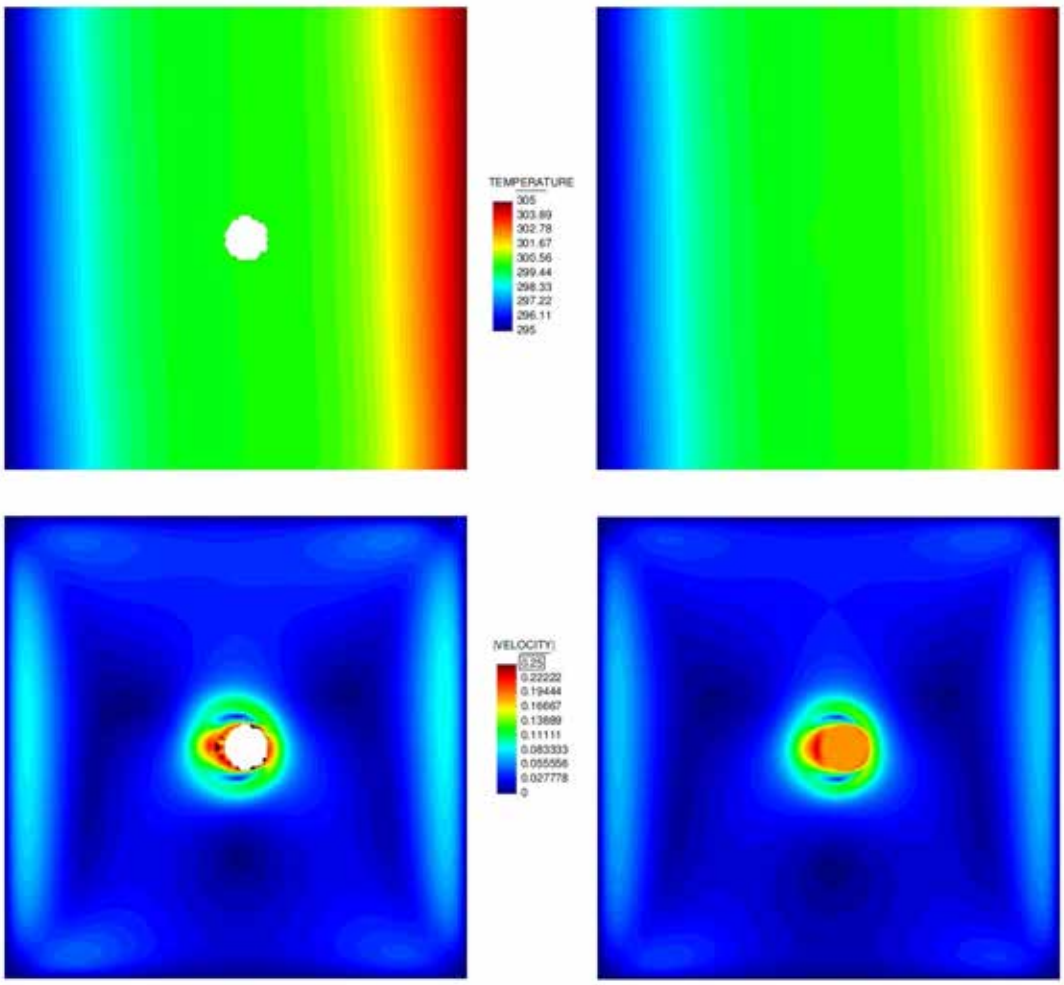


Figure 5.22: Temperature and velocity countours at t=1.4 [s]: comparison of ALE and Eulerian-Lagrangian coupling

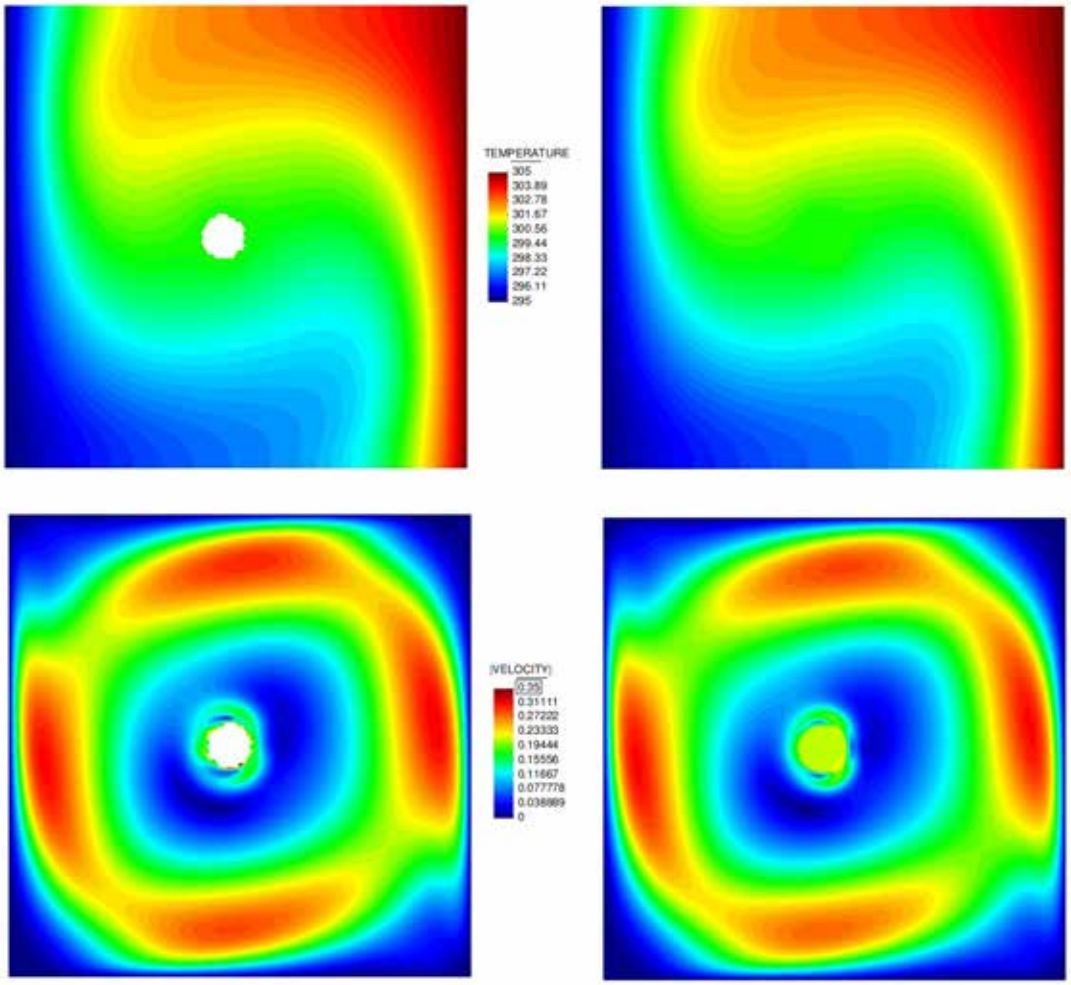


Figure 5.23: Temperature and velocity countours at t=1.4 [s]: comparison of ALE and Eulerian-Lagrangian coupling

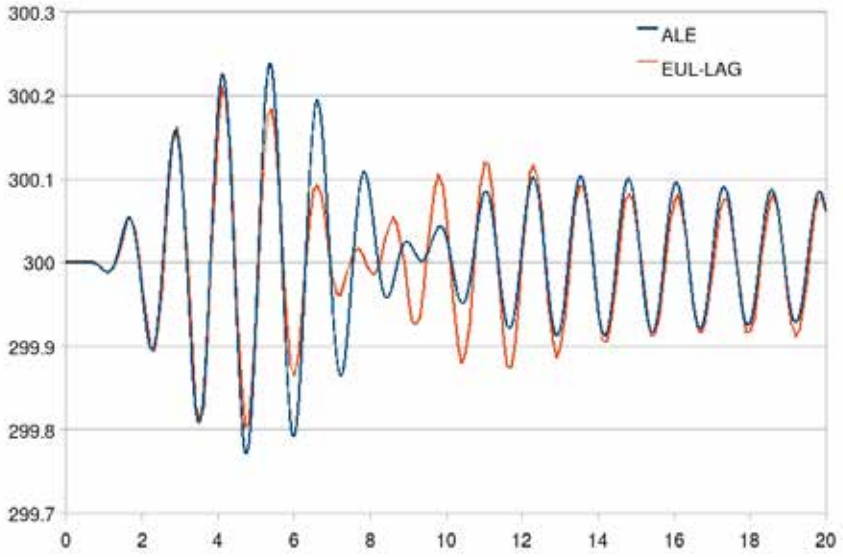


Figure 5.24: Temperature evolution at the center of cilinder: comparison of ALE and Eulerian-Lagrangian coupling

duration of the transitory stage can vary between the different solvers. The two solvers used in the present example differ both in time integration schemes and stabilization. However one can see that once the periodic state is reached, both solutions are almost coincident.

Remark In this example the comparison was carried out with the ALE formulation. Off course the possibility to treat the interaction "exactly" in the sense of the position of interface and interface boundary conditions is of great advantage. However slightly exceeding the maximum displacement in the present example leads to failure of ALE solver due to mesh deformation, while the proposed methodology allows us to treat arbitrarily large motions of the cilinder or Lagrangian object in general.

5.3.3 Polymer melting under the unsteady heat flux

In the following example we have chosen the parameters for the polymer in an artificial way, i.e. they do not represent physical values of polypropylene. The heat capacity, conductivity and initial temperature were chosen in such a way that the melting is accelerated. The intention of this example is thus exclusively to show the overall potentiality of the method.

The model A square domain of size 1m representing the room (lower left corner is located at (0,0)) contains a chair. The lower right corner of the chair is placed at (0.85,0.2), the seat and the

circular segments with the chord length of 0.3 and 0.4 [m] respectively and the radius of 0.3 [m]. The bold lines on the model represent the parts where the zero-displacement boundary condition is applied. In Fig.5.25, the Eulerian and Lagrangian model parts are presented. The boundary conditions for the Eulerian model part (the room air) are defined as follows: no-slip condition on all the walls. Heat source is located at the floor between x=0.1 and x=0.4 [m]. The heat source is modeled by prescribing the temperature of 1298 [K] at this location.

The properties of the ambient air at time t=0 are:

- density $\rho = 1.0$ $[kg/m^3]$
- kinematic viscosity ν=0.001 $[kg/m^3]$
- gravity acceleration g_y = -9.8 $[m/s^2]$
- specific isobaric heat capacity $c_p = 1000$ $J/(kg \cdot K)$
- conductivity = 1 $[W/K]$
- initial temperature $T_{init} = 393$ $[K]$

The properties of the "polymer" at time t=0 are:

- density $\rho = 100.0$ $[kg/m^3]$
- bulk modulus K = 10 E06 $[Pa]$
- kinematic viscosity ν=1.0 $[kg/m^3]$
- gravity acceleration g_y = -9.8 $[m/s^2]$
- specific isobaric heat capacity $c_p = 10$ $[J/(kg \cdot K)]$
- conductivity = 10 $[W/K]$
- initial temperature $T_{init} = 493$ $[K]$

Next we present the air temperature and respective polymer viscosity distributions at different time instances. This is shown in Fig. 5.26.

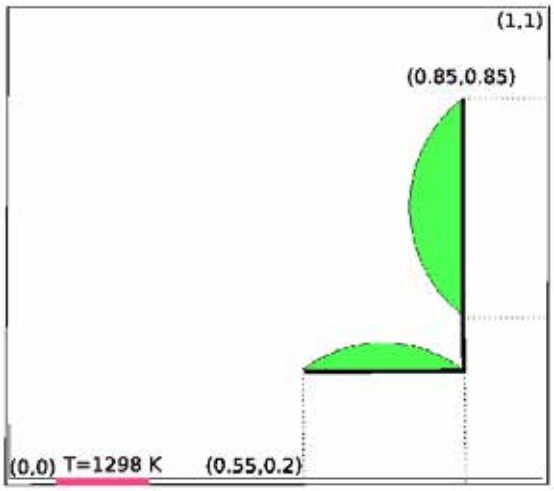
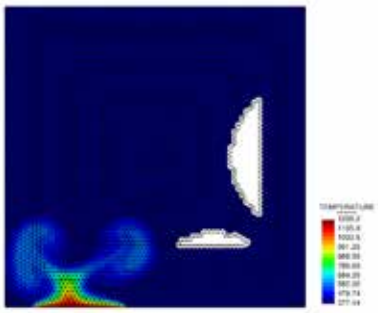
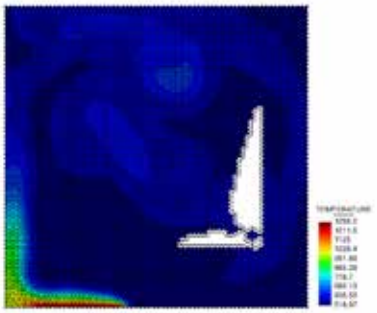


Figure 5.25: A model of polymer-made chair



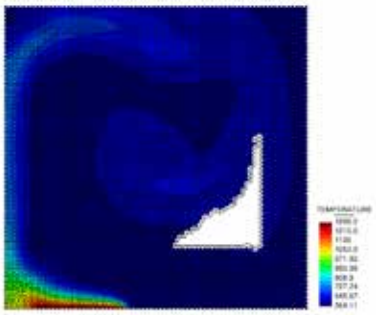
(a) Air temperature at t=1 s

(b) Polymer viscosity at t=1 s

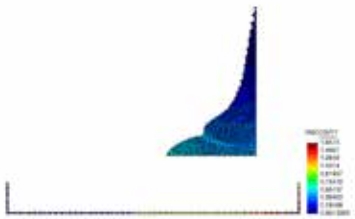


(c) Air temperature at t=18 s

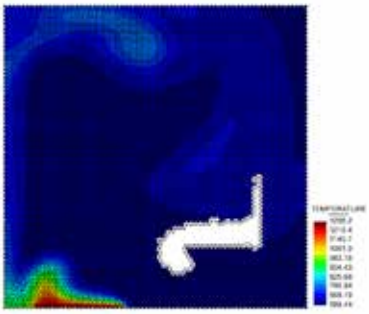
(d) Polymer viscosity at t=18 s



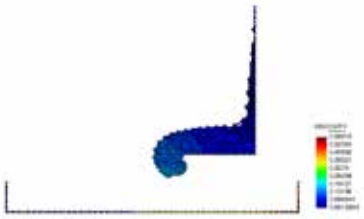
(e) Air temperature at t=22 s



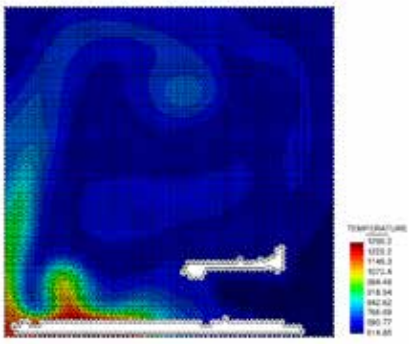
(f) Polymer viscosity at t=22 s



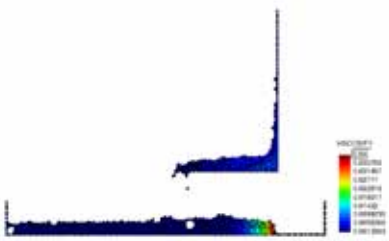
(g) Air temperature at t=25 s



(h) Polymer viscosity at t=25 s



(i) Air temperature at t=30 s

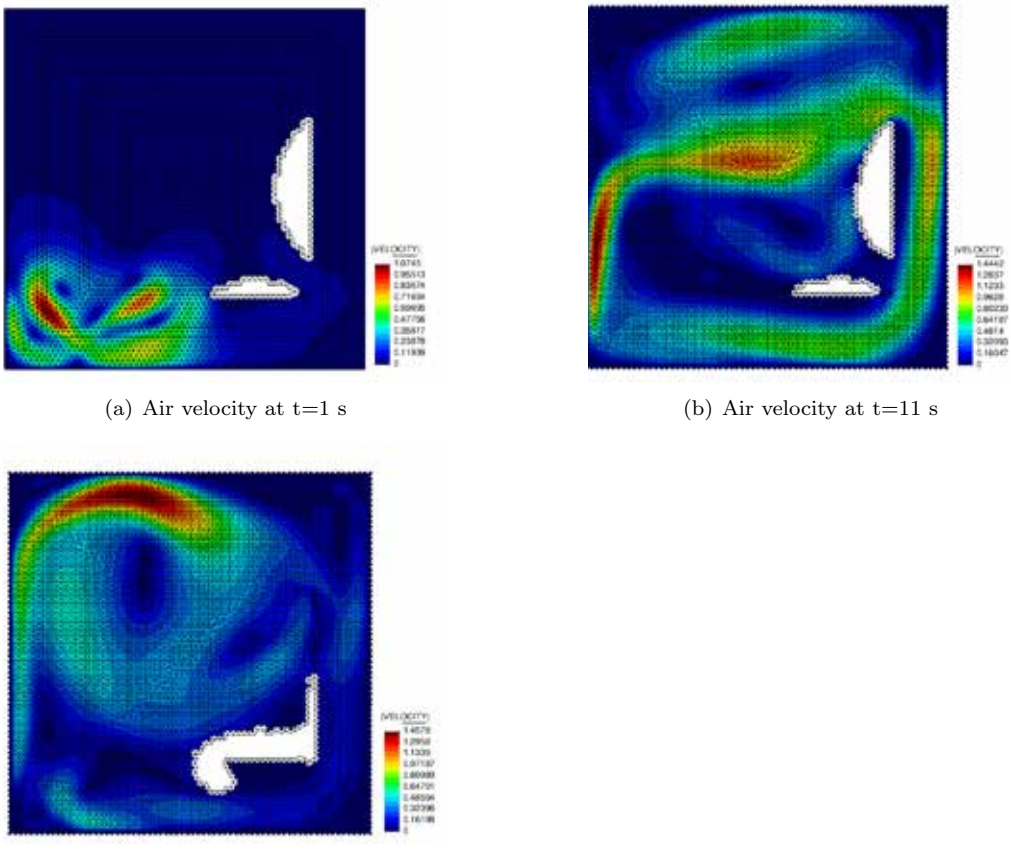


(j) Polymer viscosity at t=30 s

Figure 5.26: Air temperature and polymer viscosity at different time instances: Eulerian-Lagrangian

One can see the convective flow due to heating as well as the non-uniform distribution of temperature over the domain and the corresponding viscosity distribution of the polymer. At the early stages (up to $\propto$ 10 [s]) the viscosity of the polymer decreases, however the shape alterations are minor. We see that at later stages (e.g. 25 s) the viscosity of almost the entire polymer is low (less than 0.2) and the overall polymer motion starts. At 30[s] one can see the dripping process.

Fig. 5.27 displays the air velocity contours giving a qualitative insight of the air velocity magnitudes expectable when heating to $\propto$ 900 [C] is present. 5.26.



(a) Air velocity at t=1 s

(b) Air velocity at t=11 s

(c) Air velocity at t=25 s

Figure 5.27: Air velocity distribution at different time instances

This example presents merely potentiality of the method. Unfortunately lack of experimental data precludes the validation of the model by means of this example.

5.4 Summary and conclusions

In this chapter we have developed a framework for the solution of a complex practical problem of melting of polymer-made furniture in living areas under fire conditions. The approach was based on the PFEM model for the polymer and the Eulerian compressible model for the surrounding air. Weak one-way Dirichlet coupling was introduced. The physical justification of the simplified coupling algorithm chosen was presented.

The numerical tests performed have shown that the results obtained using the proposed approach are in good agreement with those obtained using the standard ALE method. However, while capabilities of ALE are restricted by mesh deformations, our coupled approach can tackle arbitrary deformations. On the other hand, the comparison with the monolithic Lagrangian method has also shown good agreement. The superiority of the Eulerian-Lagrangian approach for the problem of interest is the absence of a deformable background mesh and thus necessity of re-meshing exclusively a minor portion of the domain (polymer). The accuracy of the coupling strategy depends on: 1) the discretization of the interface and 2) time step size chosen(in case a staggered scheme for imposing projection Dirichlet boundary conditions is used).

Even though the model excluded such important physical phenomena as radiation and combustion, it serves as a general framework for further developments. The important feature of the method is its computational efficiency, that is achieved by modeling the ambient air (that occupies a major part of the computational domain) on a fixed grid, while modeling the polymer in the Lagrangian framework. This way the whole advantage of the Lagrangian formulation can be benefited, while the computational effort due to re-meshing is minimized. We believe that for the real practical simulations that necessitate modeling living areas of real dimensions and simulation real times over several minutes or even hours, this setting is optimal.

Chapter 6

Conclusions and future lines of research

Here we shall highlight the achievements of this work, stating the new contributions. We shall outline as well the future research lines that naturally originate from the developments of the current work.

The main objective of this work has been to find the suitable application areas of Lagrangian FE approaches for fluids and coupled problems and design corresponding algorithms. We believe that this objective has been successfully achieved. In this work we derived three qualitatively different interaction algorithms:

1. A strongly coupled partitioned FSI algorithm based on a incompressible PFEM fluid and a rigid body

2. A strongly coupled monolithic FSI algorithm based on a quasi-incompressible PFEM and a Lagrangian displacement-based solid

3. A weakly coupled algorithm for solving thermo-mechanical systems consisting of light and heavy fluids, based on Eulerian compressible and PFEM quasi-incompressible formulations

An overall conclusion is that the Lagrangian fluids are suitable for modeling free-surface flows. Also the maximum benefit can be taken from a Lagrangian fluid formulation for the fluid-structure interaction, where the symmetry of the resulting matrices (absence of convection) is advantageous for solving the FSI monolithically. The main drawback of the Lagrangian FE in fluid dynamics is the absence of control over the mesh quality. Thus several ad-hoc techniques such as adaptive time

stepping and adaptive re-meshing are mandatory. Mesh smoothing techniques typically used in ALE-PFEM formulations for fluids introduce superior mesh-quality control. However many beneficial features[1] are characteristic exclusively of the Lagrangian PFEM.

Another important conclusion is that the PFEM can be coupled easily with an Eulerian formulation, which is important for modeling the interaction between heavy and light fluids (e.g. a heavy viscous fluid and air). Tests have shown that a simple weak coupling provides accurate results.

Achievements of each chapter are summarized next:

In **Chapter 2** we have developed a computational model for an industrial problem involving simulation of sea-landing of a satellite capsule. This was done by coupling the classical PFEM formulation [1], [2] for the fluid with the rigid body model for the capsule via an improved Dirichlet-Neumann coupling. A few additional algorithms for the mesh quality improvement were introduced.

The core development of this work is the quasi-incompressible updated Lagrangian fluid formulation described in **Chapter 3**. We have shown that on the basis of this fluid formulation, strongly coupled fluid-structure interaction problems can be efficiently modeled by a monolithic Lagrangian FSI approach. An important practical advantage is the possibility of setting a monolithic system with a vast majority of existing structural codes, not necessitating any modifications in those. Optimal performance with respect to computational efficiency was attained for the fluid's bulk modulus of $\propto 10E06$ [Pa].

The convergence characteristics of the method were presented. However the detailed analysis of the linearized system conditioning in terms of the fluids' bulk modulus is an open topic and a clear line for future research. Improving the pressure stabilization (which becomes weaker for large values of bulk modulus when the methodology proposed here is used) might also be interesting for the future research in the field. Finally, the use of different preconditioners might possibly increase the limiting bulk modulus value in the sense of associated computational efficiency.

In **Chapter 4** we have developed an efficient compressible sub-sonic fluid formulation in an Eulerian framework. A semi-explicit approach was derived. We have shown that uncoupling the energy equation permits an implementation very similar to that of the incompressible case. An innovative time integration approach based on combination of a 4th order Runge-Kutta scheme with the Backward Euler method posed in the context of a fractional step split was introduced. The approach permitted to resolve efficiently the problems at low flow speeds without modifying the governing equations on the continuous level.

[1]These are: symmetry of the system, absence of stability problems in convection-dominated flows, simplicity of implementation and coupling with the structure

The novelty in **Chapter 5** consists in a new architecture for solving a complicated thermo-mechanical problem such as the melting of a polymer under fire. There, a coupling between PFEM and Eulerian fluids was established. In practice this approach defines an efficient algorithm combining the best features of Lagrangian and Eulerian descriptions. Comparison of the results obtained using the developed methodology with monolithic Lagrangian and ALE methods has proven its applicability in the area of interest.

Future work aiming at realistic representation of all complicated phenomena in a polymer melting under fire conditions in living areas will require to include combustion in the model.

The option of defining a strong two-way coupling strategy might also be appealing, first of all to qualitatively estimate the inaccuracy of the weak coupling, and second, to extend the methodology to a more general case, not restricting the analysis to a large ratio between the densities of the involved subsystems (gas and viscous polymer flow).

The aim of this work was by no means to show the general superiority of Lagrangian methods, rather we have intended to derive the strategies for a wide range of applications, where Lagrangian methodologies are natural and facilitate both the modeling and the computational solutions.

Bibliography

[1] Oñate E., Idelsohn S., Del Pin F., and Aubry R. The particle finite element method: an overview. *International Journal of Computational Methods*, 1:267–307, 2004.

[2] Idelsohn S., Oñate E., and Del Pin F. The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International Journal of Numerical Methods in Engineering*, 61:964–989, 2004.

[3] Marti J. *The particle finite element method in fluid-structure interaction.* PhD thesis, FICH, UNL, Santa Fe, Argentina, 2008.

[4] Mohamad A.A. and Viskanta R. Transient natural convection of low-prandtl-number fluids in a differentially heated cavity. *International Journal for Numerical Methods in Fluids*, 13:61–81, 1991.

[5] Principe J. *Subgrid scale stabilized finite elements for low speed flows.* PhD thesis, Technical University of Catalonia, 2008.

[6] Christon M.A. Ls-dyna and the 8:1 differentially heated cavity. *International Journal for Numerical Methods in Fluids*, 40:1133–1144, 2002.

[7] Zienkiewicz O.S., Taylor R.L., and Nithiarasu P. *The finite element method for Fluid Dynamics. 6th Edition, 3 Volumes.* Elsevier butterworthheinemann edition, 2009.

[8] Krylov A. N. Academician B.G. Galerkin: On the seventieth anniversary of his birth. *Vestnik Akademii nauk SSSR*, 4:91–94, 1941.

[9] Donea J. and Huerta A. *Finite element method for flow problems.* J. wiley edition, 2003.

[10] Brooks A.N. and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[11] T. J. R. Hughes, L.P. Franca, and G.M. Hulbert. A new finite element formulation for computational fluid dynamics: Viii. the Galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 73:173–189, 1989.

[12] Franca L.P. and T. J. R. Hughes. Convergence analyses of Galerkin least-squares methods for symmetric advective-diffusive forms of the Stokes and incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 105:285–298, 1993.

[13] T. J. R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, 73:387–401, 1995.

[14] Codina R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering*, 191:4295–4321, 2002.

[15] Osher S.J. and Fedkiw R. P. *Level Set Methods and Dynamic Implicit Surfaces*. Springer edition, 2006.

[16] Hirt C.W. and Nichols B.D. Volume of fluid (VOF) method for the dynamics of free boundaries. *Computational Physics*, 39:201–225, 1981.

[17] R.A. Gingold and J.J Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices Royal Astronomical Society*, 181:375–389, 1977.

[18] Liu G. R. *Smoothed Particle Hydrodynamics: A Meshfree Particle Method*. World scientific publishing company edition, 2003.

[19] Liu G. R. *Mesh Free Methods: Moving Beyond the Finite Element Method*. Crc press edition, 2002.

[20] Idelsohn S., Del Pin F., Rossi R., and Oñate E. Fluid-structure interaction problems with strong added-mass effect. *International Journal for Numerical Methods in Engineering*, 2009. early view.

[21] Idelsohn S., Marti J., Limache A., and Oñate E. Unified Lagrangian formulation for elastic solids and incompressible fluids. application to fluid-structure interaction problems via the pfem. *Computer Methods in Applied Mechanics and Engineering*, 197:17621776, 2008.

[22] Dadvand P., Rossi R., and Oñate E. An object-oriented environment for developing finite element codes for multi-disciplinary applications. *Archieves of Computational Methods in Engineering*. to appear.

[23] Delaunay B. Sur la sphre vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800, 1934.

[24] Akkiraju N., Edelsbrunner H., Facello M., Fu P., Mucke E. P., and Varela C. Alpha shapes: definition and software. *Proceedings of International Computational Geometry Software Workshop*, 1995.

[25] Brezzi F. and Bathe K.J. A discourse on the stability conditions for mixed finite element formulations. *Comput. Methods Appl. Mech. Engnrg*, 82:27–57, 1990.

[26] Oñate E. A stabilized finite element method for incompressible viscous. *Comp. Meth. Appl. Mech. Eng.*, 182 (1-2):355–370, 2000.

[27] Oñate E. Possibilities of finite calculus in computational mechanics. *Int. Journ. Num. Meth. Engng.*, 60 (1):255–281, 2004.

[28] Oñate E. A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation. *Computer Methods in Applied Mechanics and Engineering*, 182:355–370, 2000.

[29] Codina R. Pressure stability in fractional step finite element method for incompressible flows. *Journal of Computational Physics*, 170:112–140, 2001.

[30] Chorin A.J. A numerical method for solving incompressible viscous problems. *Journal of Computational Physics*, 2, 1967.

[31] Blair Perot J. An analysis of the fractional step method. *Journal of Computational Physics*, 108:51–58, 1993.

[32] Edelsbrunner H. and Shah N. Incremental topological flipping works for regular triangulations. *Algorithmica*, 15:223–241, 1996.

[33] Idelsohn S., Calvo N., and Oñate E. Polyhedrization of an arbitrary 3D point set. *Computer Methods in Applied Mechanics and Engineering*, 192/22-24:2649–2667, 2003.

[34] Shewchuk J. R. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22(1-3):21–74, 2002.

[35] Si H. Adaptive tetrahedral mesh generation by constrained Delaunay refinement. *Internat. J. Numer. Methods Engrg.*, 75:856–880, 2008.

[36] Rossi R, Ryzhakov P., and Oñate E. On the application of adaptive mesh refinement to the particle finite element method. First South-East European Conference on Computational Mechanics SEECCM-06, June 28-30, 2006, Kragujevac, Serbia and Montenegro, 2006.

[37] Ryzhakov P. Error estimation and adaptivity for fluid mechanics problems. Master's thesis, Technical University of Munich, TUM, 2006.

[38] Heil M., Hazel A.L., and Boyle J. Solvers for large-displacement fluidstructure interaction problems: segregated versus monolithic approaches. *Computational Mechanics*, 43:91–101, 2008.

[39] Dowell E. and Hall K. Modeling of fluid-structure interaction. *Annu. Rev. Fluid Mech.*, 33:445–90, 2001.

[40] Gerstenberger A. and Wall W. An extended finite element/lagrange multiplier based approach for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 197:1699–1714, 2008.

[41] Walhorn E., Kolke A., Hubner B., and Dinkler D. Fluid-structure coupling within monolithic model involving free surface flow. *Computers and Structures*, 83:2100–2111, 2005.

[42] Chobotov V.A. *Spacecraft Attitude Dynamics and Control.* Orbit books edition, 1991.

[43] Betsch P. and Siebert R. Rigid body dynamics in terms of quaternions: Hamiltonian formulation and conserving numerical integration. *International Journal for Numerical Methods in Engineering*, 79/4:44–473, 2009.

[44] Siebert R. and Betsch P. Numerical integration of rigid body dynamics in terms of quaternions. *Proc. Appl. Math. Mech.*, 8/1:10139–10140, 2008.

[45] Argyris J. An excursion into large rotations. *Computer Methods in Applied Mechanics and Engineering*, 32:85–155, 1982.

[46] Simo J.C. and Vu-Quoc L. On the dynamics of flexible beams under large overall motionsthe plane case. *Applied Mechanics ASME*, 53:849–863, 1986.

[47] Ibrahimbegović A., Frey F., and Kožar I. Computational aspects of vector-like parametrization of three-dimensional finite rotations. *International Journal for Numerical Methods in Engineering*, 38/21:3653–3673, 1995.

[48] Kuettler U. and Wall W. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43:61–72, 2008.

[49] Badia S., Nobile F., and Vergara C. Robin-Robin preconditioned krylov methods for fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 198/33-36:2768–2784, 2009.

[50] Houzeaux G. and Codina R. Chimera method based on a Dirichlet/Neumann(Robin) coupling for the Navier-Stokes equations. *Comp. Meth. Appl. Mech. Eng.*, 192/31-32:3343–3377, 2003.

[51] Belytschko T., Liu W.K., and Moran B. *Nonlinear finite elements for Continua and Structures.* J. wiley edition, 2003.

[52] Hübner B., Walhorn E., and Dinkler. D. A monolithic approach to fuid-structure interaction using space- time fnite elements. *Computer Methods in Applied Mechanics and Engineering*, 193:23–26, 2004.

[53] T. Hughes. *The finite element method. Linear static and dynamic FE analysis.* Dover edition, 2000.

[54] Reddy J.N. and Gartling D.K. *The finite element method in Heat Transfer and fluid dynamics.* Crc press edition, 2000.

[55] Scovazzi G., Christon M.A., Hughes T., and Shadid J. Stabilized shock hydrodynamics: I. a Lagrangian method. *Computer Methods in Applied Mechanics and Engineering*, pages 923–966, 2007.

[56] Souza Neto E.A., Peric D., Dutko M., and Owen D.R.J. Design of simple low order finite elements for large strain analysis of nearly incompressible solids. *International Journal of Solids and Structures*, 33:3277–3296, 1996.

[57] Brezzi F. and Bathe K.-J. A discourse on the stability of the mixed finite element formulations. *Journal of Computer Methods in Applied Mechanics*, 22:27–57, 1990.

[58] Reddy J.N. *An introduction to nonlinear finite element analysis*. Oxford university press edition, 2006.

[59] Rossi R., Ryzhakov P., and Oñate E. A monolithic fe formulation for the analysis of membranes in fluids. *Spatial structures*, 24(4):205–210, 2009.

[60] Codina R. An iterative penalty method for the finite element solution of the stationary navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 110:237–262, 1993.

[61] Antoci C., Gallati M., and Sibilla S. Numerical simulation of fluidstructure interaction by sph. *Journal of Computers and Structures*, 85:879–890, 2007.

[62] Alazard T. Low Mach number limit of the full NavierStokes equations. *Arch. Ration. Mech. Analysis*, 180:1–73, 2006.

[63] Chae D. and Ha S. On the formation of shocks to the compressible Euler equations. *Commun. Math. Sci.*, 7:627–634, 2009.

[64] Nakayama Y. and Boucher R. *Introduction to Fluid Mechanics*. Butterworth-heinemann edition, 1999.

[65] Gartling D.K. and Hickox C.E. A numerical study of the applicability of the boussinesq approximation for a fluid-saturated porous medium. *International Journal for Numerical Methods in Fluids*, 5/11:995–1013, 2005.

[66] Principe J. and Codina R. On the the low Mach number and the Boussinesq approximations for low speed flows. *International Journal for Numerical Methods in Fluids*. submitted.

[67] Codina R. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Comput. Methods. in Applied Mech. and Eng.*, 156:185–210, 1998.

[68] Goldberg S. and Gohberg I. *Basic operator theory*. Birkhauser boston edition, 2001.

[69] Badia S. and Codina R. Analysis of a stabilized finite element approximation of the transient convection-diffusion equation using an ale framework. *Journal on Numerical Analysis*, 44:2159–2197, 2006.

[70] Codina R., Vazquez M., and Zienkiewizc O.C. A general algorithm for compressible and incompressible flows. *International Journal for Numerical Methods in Fluids*, 27:13–32, 1998.

[71] Hauke G. and Hughes T. A unified approach to compressible and incompressible flows. *Computer Methods in Applied Mechanical Engineering*, 113:389–395, 1994.

[72] Hughes T., Franca L., and Mallet M. A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible euler and navier-stokes equations and the second law of thermodynamics. *Comp. Meth. Appl. Mech. Eng.*, 54:223–234, 1986.

[73] Hauke G. and Hughes T. A comparative study of different sets of variables for solving compressible and incompressible flows. *Computer Methods in Applied Mechanical Engineering*, 153:1–44, 1998.

[74] Aubry R., Idelsohn S. R., and Oñate E. Particle finite element method in fluid-mechanics including thermal convection-diffusion. *Computers and Structures*, 83:1459–1475, 2005.

[75] Strada M. and Heinrich J.C. Heat transfer rates in natural convection at high rayleigh numbers in rectangular enclosures: a numerical study. *Numerical Heat Transfer*, 5:81–93, 1982.

[76] Christon M.A., Gresho P.M., and Sutton S.B. Computational predictability of time-dependent natural convection flows in enclosures (including benchmark solution). *International Journal for Numerical Methods in Fluids*, 40:953–980, 2002.

[77] Gresho P. M. and Sutton S.B. Application of the fidap code to the 8:1 thermal cavity problem. *International Journal for Numerical Methods in Fluids*, 40/8:1083–1092, 2002.

[78] Xin S. and Le Quere P. An extended Chebyshev pseudo-spectral benchmark for the 8:1 differentially heated cavity. *International Journal for Numerical Methods in Fluids*, 40/8:981–998, 2002.

[79] Salinger A.G., Lehoucq R.B., Pawlowski R.P., and Shadid J.N. Understanding the 8:1 cavity via scalable stability analysis algorithms. *First MIT Conference on Computational Fluid and Solid Mechanics*, 2001. Conference proceedings.

[80] Butler K., Oñate E., Idelsohn S., and Rossi R. Modeling polymer melt flow using the Particle Finite Element Method. *International Interflam Conference, London, 11th Proceedings*, pages 929–940, 2007. Conference proceedings.

[81] Fleischmann C.M. and Hill G.R. Burning behaviour of upholstered furniture. *Interscience Communications*, pages 907–916, 2004. International Interflam Conference, London, Conference proceedings.

[82] Oñate E., Rossi R., Idelsohn S., and Bulter K. Melting and spread of polymers in fire with the particle finite element method. *International Journal for Numerical Methods*, 81/8:1046–1072, 2009.

[83] Idelsohn S. Marti J., Oñate E. Analysis of burning and melting of objects in fires situations with the particle finite element method. *Particle Methods 2009. Conference proceedings*, pages 200–207, 2009. Conference proceedings.

[84] Butler K. Model of melting and dripping thermoplastic objects in fire. *Fire and Materials 2009, San Francisco, Conference Proceedings.*, pages 341–352, 2009. Conference proceedings.

[85] Butler K., Ojlemiller T., and Linteris G. A progress report on numerical modeling of experimental polymer melt flow behavior. *Interflam 2004, Edinburgh, Scotland*, pages 937–948, 2004. International Interflam Conference, London, Conference proceedings.

[86] Peskin C.S. The immersed boundary method. *Acta Numer.*, pages 479–517, 2002.

[87] de Berg M., van Krevald M., Overmars M., and Schwarzkopf O. *Computational Geometry Algorithms and Applications.* Springer-verlag edition, 1997.

[88] R. Codina, G. Houzeaux, H. Coppola-Owen, and J. Baiges. The fixed-mesh ALE approach for the numerical approximation of flows in moving domains. *Journal of Computational Physics archive*, 228:1591–1611, 2009.